UNIVERSITY OF CALIFORNIA SAN DIEGO

Modeling the Dynamics of Consumer Behavior from Massive Interaction Data

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Computer Science

by

Mengting Wan

Committee in charge:

Professor Julian McAuley, Chair
Professor Taylor Berg-Kirkpatrick
Professor Kamalika Chaudhuri
Professor Virginia De Sa
Professor Ndapandula Nakashole

2019

The dissertation of Mengting Wan is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____

Chair

University of California San Diego

2019

DEDICATION

To my family.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I am sincerely grateful to my advisor Professor Julian McAuley who carefully protected my curiosity and research interests, and generously guided me throughout this journey. Many thanks to Professors Taylor Berg-Kirkpatrick, Kamalika Chaudhuri, Ndapa Nakashole and Virginia De Sa for serving on my thesis committee and offering their time and insightful comments. I am also indebted to Professor Jiawei Han who introduced me into the data mining research area and helped me along the way. I appreciate all my collaborators, labmates and colleagues: Dr. Di Wang, Dr. Jie Liu, Prof. Ndapa Nakashole, Dr. Dimitrios Lymberopoulos, Dr. Matt Goldman, Dr. Matt Taddy, Dr. Paul Bennett, Dr. Justin Rao, Dr. Cindy Chen, Rishabh Misra, Wang-Cheng Kang, Jianmo Ni, and many others. Special thanks to my mentors Di Wang at Microsoft Research and Cindy Chen at Airbnb, for their encouragement and valuable suggestions about my career.

I feel extremely fortunate that I was raised by my amazing parents with their deepest love and support. Thanks to all my friends at UC San Diego: Sai Bi, Zexiang Xu, Lifan Wu, Yao Qin, Zhen Zhai, Songbai Yan, Yizhen Wang, Julaiti Alafate, etc. Many thanks to Shuai Tang, Wang-Cheng Kang, Jianmo Ni for their inputs and open-mindedness about research and life. Thank you Clare Yi Xu, Ran He, Fan Hao and many other my beloved friends across the world for being a source of my cheers and tears and demonstrating to me how fantastic girls can be.

Chapter 3, is based on the materials as they appear in the *International Conference on World Wide Web*, 2017 ("Modeling Consumer Preferences and Price Sensitivities from Large-Scale Grocery Shopping Transaction Logs," Mengting Wan, Di Wang, Matt Goldman, Matt Taddy, Justin Rao, Jie Liu, Dimitrios Lymberopoulos, and Julian McAuley), and the *ACM Conference on Information and Knowledge Management*, 2018 ("Representing and Recommending Shopping Baskets with Complementarity, Compatibility, and Loyalty," Mengting Wan, Di Wang, Jie Liu, Paul Bennett, and Julian McAuley). The dissertation author was the primary investigator and author of these papers.

Chapter 4, contains the material as it appears in the *ACM Conference on Recommender*

Systems, 2018 ("Item Recommendation on Monotonic Behavior Chains," Mengting Wan and Julian McAuley). The dissertation author was the primary investigator and author of this paper.

Chapter 5, is based on the materials as they appear in the *IEEE International Conference on Data Mining*, 2016 ("Modeling Ambiguity, Subjectivity, and Diverging Viewpoints in Opinion Question Answering Systems," Mengting Wan and Julian McAuley), and the *SIAM International Conference on Data Mining*, 2018 ("One-Class Recommendation With Asymmetric Textual Feedback," Mengting Wan and Julian McAuley). The dissertation author was the primary investigator and author of these papers.

Chapter 6, contains the material to appear in *ACM Conference on Web Search and Data Mining*, 2020 ("Addressing Marketing Bias in Product Recommendations," Mengting Wan, Jianmo Ni, Rishabh Misra, and Julian McAuley). The dissertation author was the primary investigator and author of this paper.

VITA

2009–2013	B. S. in Statistics, Peking University
2013–2015	M. S. in Statistics, University of Illinois Urbana-Champaign
2015–2019	Ph. D. in Computer Science, University of California San Diego

PUBLICATIONS

Mengting Wan, Jianmo Ni, Rishabh Misra, Julian McAuley, "Addressing Marketing Bias in Product Recommendations", to appear in Proceedings of *2020 ACM Conference on Web Search and Data Mining* (WSDM), 2020.

An Yan, Shuo Cheng, Wang-Cheng Kang, **Mengting Wan**, Julian McAuley, "2D Convolutional Neural Networks for Sequential Recommendation", in Proceedings of *2019 ACM Conference on Information and Knowledge Management* (CIKM), 2019.

Mengting Wan, Rishabh Misra, Ndapa Nakashole, Julian McAuley, "Fine-Grained Spoiler Detection from Large-Scale Review Corpora", in Proceedings of *2019 Annual Meeting of the Association for Computational Linguistics* (ACL), 2019.

Mengting Wan, Cindy Chen, "Beyond 'How may I help you?': Assisting Customer Service Agents with Proactive Responses", in Proceedings of *2019 AAAI Workshop on Reasoning and Learning for Human-Machine Dialogues* (DEEP-DIAL), 2019.

Mengting Wan, Di Wang, Jie Liu, Paul Bennett, Julian McAuley, "Representing and Recommending Shopping Baskets with Complementarity, Compatibility, and Loyalty", in Proceedings of *2018 ACM Conference on Information and Knowledge Management* (CIKM), 2018.

Wang-Cheng Kang, **Mengting Wan**, Julian McAuley, "Recommendation Through Mixtures of Heterogeneous Item Relationships", in Proceedings of *2018 ACM Conference on Information and Knowledge Management* (CIKM), 2018.

Mengting Wan, Julian McAuley, "Item Recommendation on Monotonic Behavior Chains", in Proceedings of *2018 ACM Conference on Recommender Systems* (RecSys), 2018.

Rishabh Misra, **Mengting Wan**, Julian McAuley, "Decomposing Fit Semantics for Product Size Recommendation in Metric Spaces", in Proceedings of *2018 ACM Conference on Recommender Systems* (RecSys), 2018.

Mengting Wan, Julian McAuley, "One-Class Recommendation With Asymmetric Textual Feedback", in Proceedings of *2018 SIAM International Conference on Data Mining* (SDM), 2018.

Mengting Wan, Di Wang, Matt Goldman, Matt Taddy, Justin Rao, Jie Liu, Dimitrios Lymberopoulos, Julian McAuley, "Modeling Consumer Preferences and Price Sensitivities from Large-Scale Grocery Shopping Transaction Logs", in Proceedings of *2017 International Conference on World Wide Web* (WWW), 2017.

Mengting Wan, Julian McAuley, "Modeling Ambiguity, Subjectivity, and Diverging Viewpoints in Opinion Question Answering Systems", in Proceedings of *2016 IEEE International Conference on Data Mining* (ICDM), 2016.

Mengting Wan, Xiangyu Chen, Lance Kaplan, Jiawei Han, Jing Gao, Bo Zhao, "From Truth Discovery to Trustworthy Opinion Discovery: An Uncertainty-Aware Quantitative Modeling Approach", in Proceedings of *2016 ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (KDD), 2016.

Mengting Wan, Yunbo Ouyang, Lance Kaplan, Jiawei Han, "Graph Regularized Meta-path Based Transductive Regression in Heterogeneous Information Network", in Proceedings of *2015 SIAM International Conference on Data Mining* (SDM), 2015.

ABSTRACT OF THE DISSERTATION

Modeling the Dynamics of Consumer Behavior from Massive Interaction Data

by

Mengting Wan

Doctor of Philosophy in Computer Science

University of California San Diego, 2019

Professor Julian McAuley, Chair

Recent technological innovations (e.g. e-commerce platforms, automated retail stores) have enabled dramatic changes in people's shopping experiences, as well as the accessibility to incredible volumes of consumer-product interaction data. As a result, machine learning (ML) systems can be widely developed to help people navigate relevant information and make decisions. Traditional ML systems have achieved great success on various well-defined problems such as speech recognition and facial recognition. Unlike these tasks where datasets and objectives are clearly benchmarked, modeling consumer behavior can be rather complicated; for example, consumer activities can be affected by real-time shopping contexts, collected interaction data can be noisy and biased, interests from multiple parties (both consumers and producers) can be

involved in the predictive objectives.

The primary goal of this dissertation is to address the obstacles in modeling consumer activities through computational approaches, but with careful considerations from *economic* and *societal* perspectives. Intellectually, such models help us to understand the forces that guide consumer behavior. Methodologically, I build algorithms capable of processing massive interaction datasets by connecting well-developed ML techniques and well-established economic theories. Practically, my work has applications ranging from recommender systems, e-commerce and business intelligence.

Chapter 1

Introduction

Understanding and characterizing consumer behavior, i.e., activities associated with purchases and consumptions of commodities, has been a major theme in the areas of economics and marketing throughout history. From classical microeconomics to behavioral economics, researchers seek to study consumer activities from different perspectives via different approaches (e.g. surveys, interviews, field trails, mathematical and statistical models). With the massive available data volume in the contemporary age, computer scientists stepped into the study of consumer behavior. A notable early breakthrough comes from the association rule learning for market analysis, which is a rule-based machine learning (ML) approach to uncover the co-existence patterns between products within the same shopping baskets in large databases [7, 8]. Later the innovation of e-commerce enabled the possibility that consumers' activities can be intervened through personalized recommender systems. As a result, collaborative filtering algorithms, as a group of ML methods, have greatly succeeded in connecting consumers to relevant products across the numerous candidates on e-commerce applications (e.g. Amazon [100]) and streaming services (e.g. Netflix [90], Youtube [35]).

Despite the explosive growth of machine learning research, the acceptance of ML methods has been relatively slower in economics and marketing compared to the broader quantitative

community [12]. A possible explanation is that most ML methods neither naturally provide intuitive interpretations of the consumer behavior mechanism nor deliver theoretical guarantees (which is highly valued in the econometrics community). On the other hand, economic theories and econometric methods have barely been explored in the applied ML community for cutting-edge technological applications, possibly due to their rigorous statistical assumptions and limited extendibility in the big data settings. My research therefore seeks to connect state-of-the-art ML techniques, classic and modern economic theories, as well as abundant heterogeneous consumer-product interaction data. Methodologically, materials in this dissertation still fall under the umbrella of ML frameworks. We preserve the essence of desired economic properties in the model design but evaluate the model performance in terms of the out-of-sample predictive or generalization power.

Compared to traditional econometric methods, the assumptions of the statistical structures in our models are largely relaxed to facilitate the applicability to large-scale datasets. Apart from the single 'purchase' action, multiple types of consumer behavior including structured activities (e.g. click, add-to-cart) and unstructured texts (e.g. reviews, comments) can be incorporated.

Compared to traditional ML studies, especially methods developed for recommender systems, we reconsider the model *assumptions* by associating with empirical economic observations, redefine the model *objectives* by leveraging established marketing theories, and address the potential *societal concerns* of ML systems such as marketing bias and recommendation fairness.

1.1 Overview of Contributions

This dissertation presents research studies about how to model the dynamics of consumer behavior from large-scale interaction data. We introduce a brief background in Chapter 2. The subsequent chapters are organized as follows.

In Chapter 3, we present a series of works which aims to embed several *economic factors*,

namely price, complementarity and loyalty, into product representations and recommendations. We propose a personalized, interpretable and scalable framework **NFMF**, which is capable of providing satisfying product recommendations and *price sensitivity* estimations (i.e., consumers' reactions to a price drop). We propose a product representation learning model called **triple2vec** to capture the consumer-product preference compatibility and the product-product *complementarity* simultaneously. We propose an algorithm called **adaLoyal**, which adaptively balances users *product loyalty* memorized from their historical activities and their long-term tastes inferred from the low-dimensional representations. Experiments on real-world datasets show that the price-aware model **NFMF**, the complementarity-aware model **triple2vec** and the loyalty-aware model **adaLoyal** not only yield outstanding recommendation performance but also deliver reasonable economic interpretations and insights.

In Chapter 4, we focus on the potential *structures* behind heterogeneous consumer actions (e.g. click, purchase, review). We investigate different types of consumer feedback signals and observe the monotonic dependency structures, i.e., any signal necessarily implies the presence of a weaker (or more implicit) signal (a review action implies a purchase action, which implies a click action, etc.). We therefore develop an algorithm which effectively models multiple types of interactions and efficiently exploits the structure among these actions. We evaluate the model on diverse real-world datasets where the effectiveness of the proposed algorithm is quantitatively demonstrated.

In Chapter 5, we investigate how *unstructured* texts (e.g. product reviews, product Q/As) can be utilized to address consumer behavior. We propose a Bayesian ranking method called **PRAST** which leverages consumers' asymmetrical textual feedback (e.g. review texts, comments) to facilitate product recommendations. Beyond the generic preference-matching, we propose a learning-based QA system which retrieves relevant information from consumers' textual feedback to address complex product-related questions, with emphasis on their potential ambiguity and subjectivity.

In Chapter 6, we critically discuss the potential *marketing bias* (e.g. different selection strategies of human model in a product image may result in users' different reactions) in modeling consumer behavior. We investigate the correlation between users interaction feedback and products marketing images on real-world e-commerce datasets, examine the response of several standard ML-based recommendation algorithms to the potential bias in the data, then propose a fairness-aware framework to mitigate this potential marketing bias in product recommendations.

Chapter 2

Background

A major application of consumer behavior models described in this dissertation is *recommender system*, which broadly aims to model people's preferences, opinions and behavior. In this chapter, we first introduce the basics of ML model-based recommendation methods as well as different types of consumer feedback. Then we discuss the potential problems of existing studies and how these challenges are addressed in the subsequent chapters of this dissertation.

2.1 Model-Based Recommender Systems

By matching users to the relevant information, recommender systems are already widely deployed in e-commerce platforms and intelligent brick-and-mortar retail stores. From the algorithmic perspective, we seek to build ML models to collect and generalize patterns from users historical feedback signals, i.e. the *collaborative filtering* principle. These signals generally come in *explicit structured* feedback (e.g. star-ratings), *implicit structured* feedback (e.g. clicks) and *unstructured textual* feedback (e.g. review texts).

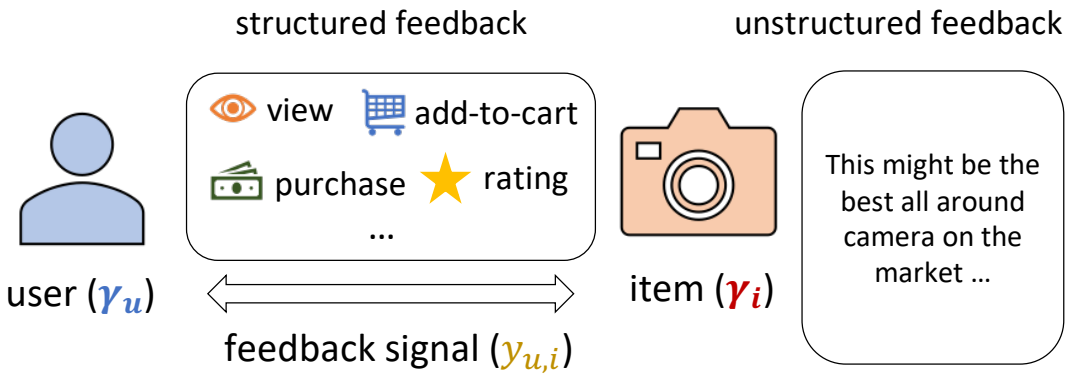


Figure 2.1: An illustration of structured and unstructured feedback signals in typical recommender systems.

2.1.1 Structured Feedback

Most recommender systems aim to model *structured* consumer feedback such as numeric rating scores and binary purchase actions, which are abundant, easy to process, and directly or indirectly reflecting consumers' inclinations. These feedback signals are normally categorized as the following two types.

- **Explicit Feedback.** Recommendation algorithms historically focused on studying users' *explicit* feedback, where users' preferences are directly exhibited through signals such as star-ratings and thumbs ups/downs [86, 87, 90]. As in the well-known Netflix Prize Challenge[1], the predictive task in the explicit feedback setting is often formulated as a rating prediction task, where products with higher predicted rating scores are recommended to a consumer. Although such feedback signals explicitly reflect if consumers like/unlike the products, these interactions are relatively scarce in terms of quantity.
- **Implicit Feedback.** On the other hand, *implicit* feedback (e.g. purchases, views, check-ins) is widely available in modern information systems, where users reveal their preferences through actions. Many algorithms have been recently proposed to model the dynamics of

[1] https://www.netflixprize.com/

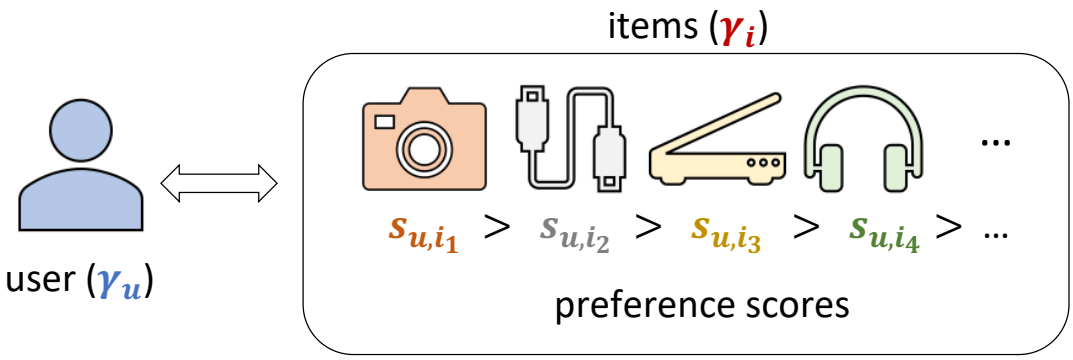


Figure 2.2: A typical recommendation algorithm is trained on consumers' feedback signals and generate preference scores to approximate these signals. The outcome from a recommender system is a list of items which are ranked based on the predicted preference scores.

these consumer activities thus to improve the recommendation performance [72, 123, 139]. In implicit-feedback settings, the predictive task is normally formulated as a *pointwise* action prediction task (i.e. predicting the presence of an interaction) or a pairwise ranking task (i.e. ranking an interacted product higher than an non-interacted product).

Latent Factor Models

On account of the Netflix Prize Challenge, latent factor models as variants of matrix factorization (MF) become popular choices to approximate the underlying mechanism behind the structured user-item interactions. The principle of latent factor models is using the inner product between latent low-dimensional user and item vectors to model the compatibility between users and items [90]. Specifically a user u's preference score towards a product item i can be calculated as

$$s_{u,i} = b_0 + b_u + b_i + \langle \boldsymbol{\gamma}_u, \boldsymbol{\gamma}_i \rangle , \tag{2.1}$$

where $b_0, b_i, b_u, \boldsymbol{\gamma}_i, \boldsymbol{\gamma}_u$ are learnable parameters. b_0 is the global offset, b_i, b_u are item and user biases, and $\boldsymbol{\gamma}_i, \boldsymbol{\gamma}_u$ are K-dimensional embeddings to capture items' latent features and users' latent preferences toward these features. K is a pre-defined hyper-parameter and is much smaller than

the number of products and users. As showed in Figure 2.2, product items are ranked based on the predicted scores $s_{u,i}$ and highly preferred product items are recommended to the user u.

In the *explicit* feedback settings (e.g. rating prediction), a common loss function is based on the Mean Squared Error (MSE) [86, 87, 90]

$$\min \sum (r_{u,i} - s_{u,i})^2, \tag{2.2}$$

where the preference score $s_{u,i}$ is used to estimate user u's rating $r_{u,i}$ on item i in the training data.

In the *implicit* feedback settings, a weighted *pointwise* loss function can be applied to approximate users' interactions. Two typical forms include the MSE-based loss [72, 123]

$$\min \sum w_{u,i}(y_{u,i} - s_{u,i})^2, \tag{2.3}$$

and the cross-entropy-based loss [67, 154]

$$\min \sum w_{u,i}\Big(y_{u,i} \log \sigma(s_{u,i}) + (1 - y_{u,i}) \log(1 - \sigma(s_{u,i}))\Big), \tag{2.4}$$

where $y_{u,i} \in \{0, 1\}$ is the interaction signal (e.g. $y_{u,i} = 1$ indicates product i is purchased by user u), $\sigma(\cdot)$ is the sigmoid function (i.e., $\sigma(x) = \frac{1}{1+\exp(-x)}$). Note in these settings, we typically observe much less positive signals (the presence of actions) compared with unobserved interactions. Therefore $\{w_{u,i}\}$ is introduced as a set of customized weights to balance positive and negative instances.

Given the positive itemset I_u^+ (e.g. products purchased by the user u) and the 'negative itemset' I_u^- for a given user, another popular objective function is a *pairwise* ranking loss [139, 140], which seeks to maximize a pairwise difference between observed purchases and

unobserved 'negative' products:

$$\min \quad - \sum_{u, i^+ \in I_u^+, i^- \in I_u^-} \log \sigma(s_{u,i^+} - s_{u,i^-}). \tag{2.5}$$

The latent factors models (Eq. (2.1)) now becomes the basis of many state-of-the-art recommendation algorithms [32, 36, 65–67, 157, 180]. By applying the state-of-the-art deep neural networks, the latent item ($\boldsymbol{\gamma_i}$) and user ($\boldsymbol{\gamma_u}$) factors can be extended to capture rich side information such as video [36], visual [65,66] and relational [160,169,178] signals. These models have been sucessfully deployed in real-world commercial applications including YouTube [36], Google Play [36], Pinterest [169] and Alibaba [180].

2.1.2 Unstructured Feedback

In addition to the structured consumer feedback signals, *unstructured* textual information (e.g. Amazon reviews, Youtube comments) may provide rich context to better predict or explain users' actions. With the fast growth of the language models in the area of natural language processing (NLP), modeling unstructured texts along with consumer actions has become an emerging research topic in the recommender system community [28, 43, 101, 111, 113, 120, 179]

One notable progress in this area is incorporating users' review texts to improve recommendation results in explicit settings (i.e., rating predictions) [28, 43, 101, 111, 150, 179]. Several such models leverage topic models such as variants of Latent Dirichlet Allocation (LDA) [22] to link the latent topics discovered in text and users' latent preference dimensions ($\boldsymbol{\gamma_u}$ in Eq. (2.1)), including **HFT** ('Hidden Factors and Topics') [111], **JMARS** ('Jointly Modeling Aspects, Ratings, and Sentiments') [43], **RMR** ('Ratings Meet Reviews') [101], **FLAME** ('Factorized Latent Aspect Model') [163] and **SLUM** ('Sentiment Utility Logistic Model') [18]. Inspired by the great success of deep neural network approaches in NLP, deep learning architectures are later developed to capture more expressive semantics of items from review texts, and to model more complex

relationships between product semantics and users' preference dimensions [28, 144, 150, 179]. **DeepCoNN** ('Deep Cooperative Neural Networks') [179] and **TransNet** ('Transformational Neural Networks') [28] learn latent representations from review text through Convolutional Neural Networks (CNNs); **D-Attn** ('Dual Attention-Based Model') [144] and **MPCN** ('Multi-Pointer Co-Attention Network') [150] utilize the co-attention mechanism to combine different views (structured rating scores and unstructured review texts) of user-item interactions. Extensive experiments have demonstrated that these textual reviews are helpful in terms of recommendation performance, particularly on 'cold' items where few interactions are included in the training data.

2.1.3 Challenges and Opportunities

We approach the limitations of existing ML systems for consumer activities from three aspects: (1) the lack of economic interpretations; (2) the lack of holistic view to model heterogeneous consumer feedback signals; (3) the potential marketing bias in the interaction data and machine learning systems. We review these challenges and show how these obstacles are addressed in this dissertation.

Economic Interpretations

A major challenge of latent factor models and their variants is that the captured dynamics are relatively difficult to interpret from the economic perspective. These ML models could be powerful in terms of predicting consumers' future behavior, but may fall short of delivering useful messages to help different sides of the market (e.g. both consumers and product retailers/producers) make decisions. In Chapter 3, we address several economic factors (i.e. price, complementarity and loyalty) on top of standard ML models. We show that by considering well-established economic theories and observations, the recommendation performance can be advanced. More importantly, many economic insights can be naturally provided so that practitioners are able to precisely react to the market dynamics.

Compared to typical economic studies which are generally in pursuit of validating proposed theories through qualitative (e.g. interviews) or quantitative (e.g. data analysis) approaches, studies in this dissertation focus on examining and incorporating these theories in ML algorithm designs. Detailed concepts and techniques from economics and marketing will be introduced in the subsequent chapters along with our proposed algorithms.

Heterogeneity of Interaction Data

We observe that two paradigms for structured interaction feedback—explicit and implicit feedback—have long been studied as two separate problems so that their distinct properties are addressed by different techniques. However, multiple heterogeneous types of feedback signals are often available in real-world scenarios (e.g. clicks, add-to-cart, purchases are available on e-commerce platforms) while the structures behind these signals are typically ignored in previous model designs. In Chapter 4, we therefore address the gap between explicit and implicit feedback and explore the *monotonic* structure behind a spectrum of users responses.

Although review texts have proven helpful when explaining and predicting explicit feedback, textual feedback has been rarely studied in implicit-feedback settings. By definition textual feedback signals such as product comments and tips without rating scores are only available for positive feedback instances (e.g. review text is never available for products a user hasn't purchased). In addition to standard review feedback, community question answering systems are widely available on e-commerce platforms as well, where consumers are allowed to post and answer specific product-related questions such as "Is this a good lens for my Nikon D3300 camera?". In Chapter 5, we first extend the existing work and model unstructured and asymmetric textual feedback in implicit-feedback scenarios. Beyond the simple preference-matching as in conventional recommender systems, we show that textual feedback is also a valuable resource to address unstructured, complex, subjective and potentially ambiguous product queries.

Bias and Fairness of Behavioral Models

In Chapter 6, we move beyond the recommendation accuracy and start investigating the recommendation fairness across the entire market. Similar to the latent factor models, many modern recommendation algorithms seek to provide personalized product recommendations by uncovering patterns in consumer-product interactions. However, these interactions can be biased by how the product is marketed, for example due to the selection of a particular human model in a product image. These correlations may result in the underrepresentation of particular niche markets in the interaction data; for example, a female user who would potentially like motorcycle products may be less likely to interact with them if they are promoted using stereotypically 'male' images. We investigate the above marketing bias on real-world datasets and develop a fairness-aware framework to mitigate this potential bias from the algorithmic perspective.

Chapter 3

Modeling Economic Factors in Consumer Activities

3.1 Introduction

Product preferences are reflected by purchase incidence or purchase quantity in a consumer's shopping history. From item-based collaborative filtering [141] to matrix factorization techniques [90], various methods for consumer preference matching have been developed in the field of recommender systems. However, there are few studies where economic factors such as product price, product complementarity and brand loyalty, let alone the relationship between consumer preferences and these factors. In this chapter, we target to build frameworks to connect well-developed techniques in recommender systems and well-established behavioral economic theories.

Overview of the Chapter

In order to match shoppers with desired products and provide personalized promotions, whether in online or offline shopping worlds, it is critical to model both consumer preferences

and **product price sensitivities** simultaneously. Personalized preferences have been thoroughly studied in the field of recommender systems, though price (and price sensitivity) has received relatively little attention. At the same time, price sensitivity has been richly explored in the area of economics, but typically not in the context of developing scalable, working systems to generate recommendations. Thus in Section 3.2, we discuss how to model consumer preferences and price sensitivities at scale, bridging a gap between modern recommender systems and economic demanding systems.

A **complementary product** is a commodity or service used in conjunction with another product. For example, shoppers may buy plastic cups and beer together for a party. In terms of economics, the price decrease of either product results in the increase of demand for both products [27]. In Section 3.3, we discuss how this product complementarity can be encoded within low-dimensional product representation vectors in ML frameworks.

Product and brand loyalty is another notable pattern in consumer behavior, especially in consumers' day-to-day grocery shopping trips [38, 73, 134]. That is users generally tend to exhibit strong affinities towards certain products, i.e., repeatedly purchasing the same product over time. Such behavior is often contrary to the low-rank assumption of interaction data implicated in latent factor models, but can be 'memorized' based on simple statistics such as purchase frequency. Motivated by this observation, we propose an algorithm in Section 3.4 which adaptively balances the 'must-buy' products with the preferences inferred from the conventional low-dimensional representations.

3.2 Price Sensitivity

Our goal is to model consumer preferences and price sensitivities from transaction data, in order to support scalable recommendation and demand-forecasting systems.

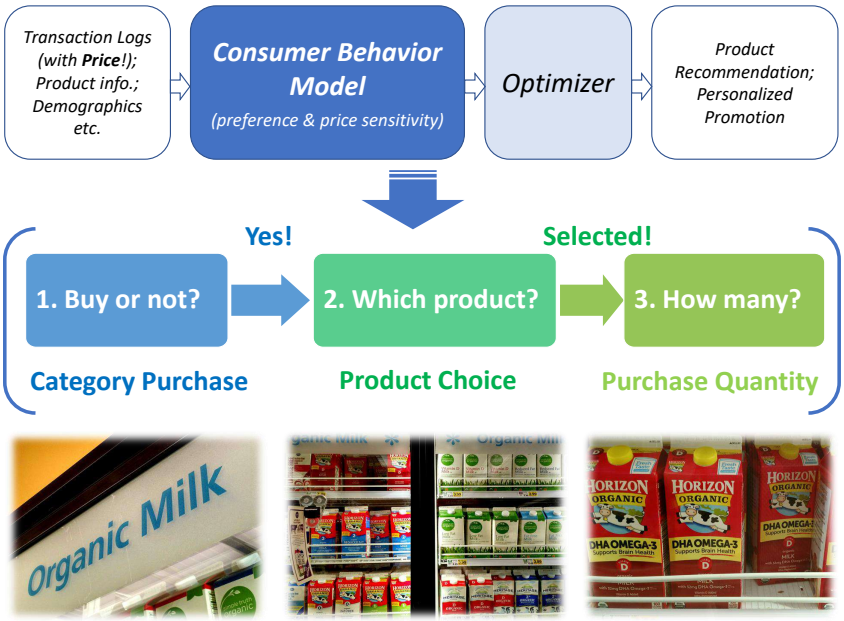


Figure 3.1: General workflow of the proposed three-stage purchase decision model.

Recommender & Demand Systems.

The general workflow of the type of hybrid, large-scale recommendation and demand system we are considering is shown in Figure 3.1. We feed large transaction logs including product prices, meta-data, and consumer information to our behavioral model which generates feedback in the form of purchase predictions. On top of this model, we apply different optimization rules to provide user-specific results. For example, personalized ranked lists can be provided by matching preferences, or customized promotion strategies can be provided based on estimated price elasticity. Or hybrid personalized coupon lists can be provided by combining preference-matching and price-matching criteria. To achieve these goals, we need to consider consumer preferences in concert with price sensitivities in our behavioral model.

Preferences & Price Sensitivities.

Product preferences are reflected by purchase incidence or purchase quantity in a consumer's shopping history. From item-based collaborative filtering [141] to matrix factorization

techniques [90], various methods for consumer preference matching have been developed in the field of recommender systems. However, there are few studies where price is considered as a factor. On the other hand, price sensitivity has been richly studied in the areas of economics and marketing, from classic demand systems [39] to customized promotion models [172, 173]. Demand systems are used to explore the relationship between product prices and quantities sold. In this context, price sensitivity is measured by the *'price elasticity'* value obtained from a demand system, which is defined as the unit change of purchase quantity (or probability) given a unit fluctuation in price [108]. In practice, elasticity-based consumer segments are considered and separate demand models are constructed for different consumer segments. Such segments can be regarded as useful signals for retailers and manufacturers to identify consumer groups to target. However, there are two limitations in current demand systems: (1) the data volumes involved are typically limited in terms of the number of products, categories and shopping trips [1] and (2) classic demand models are not able to be updated efficiently.

Therefore the major goal of this study is to construct an *interpretable* framework to model consumer preferences and price sensitivities *at scale*, by connecting large-scale recommender systems and established consumer theories from demanding systems in economics.

Three-Stage Purchase Decision Model.

Different from modeling user preferences as a whole process, we follow the three-stage framework from recent customized promotion studies [172, 173]. We notice that in real-world grocery shopping scenarios, products can be categorized either based on an existing commodity hierarchy or by clustering their associated characteristics (e.g. text descriptions). Each category should consist of some kind of products where consumers' purchase decisions share similar patterns. For example, one category might be 'organic milk' and two products in this category could be 'horizon organic whole milk' and 'organic valley whole milk.' As shown in Figure 3.1,

[1] Typical demand system studies [39, 60, 172, 173] usually involve only several products, categories and several hundred transactions.

we assume that for a given category, consumers' purchase decisions can be decomposed into three stages: (1) category purchase incidence, (2) product choice, and (3) purchase quantity. In a complete purchase decision-making process, stages are heterogeneous and consumers may behave quite differently across them. Given the fact that there are more than ten thousand distinct products in a typical grocery store [2], this three-stage model is more efficient compared with a flat model without fine-grained product categorization [5, 40], since it can be constructed in parallel across product categories and explicitly interpreted across different purchase stages.

Specifically, we first consider whether a consumer will make a purchase from a particular category in a certain shopping trip,[2] which can be regarded as a binary prediction problem. If so, we model their purchase from this category following a multinomial distribution. Third, we determine what quantity of the product will be purchased, which leads to a numeric prediction problem. This combination of binary, categorical, and numeric prediction is quite different from that used by traditional recommender systems, requiring new approaches to be developed. In particular, we develop a nested framework and extend state-of-the-art feature-based matrix factorization models to include price as a factor; this framework is embedded in the above prediction tasks with different link functions. We evaluate our model on two real-world grocery shopping datasets where our experiments reveal that the proposed framework is capable of providing high-quality preference predictions and personalized price sensitivity estimates.

Summary of Contributions

In this study, we model consumer preferences and price sensitivities for grocery shopping scenarios at scale, bridging a gap between large-scale recommender systems and established economic theories.

We propose a nested feature-based matrix factorization framework, which is flexible enough to include a range of features, to fit different prediction scenarios (for different stages

[2] A 'shopping trip' in this study is represented as a (consumer, timestamp) pair.

of purchase behavior), to be applied with scalable learning algorithms (e.g. stochastic gradient descent) and can be updated efficiently.

By applying matrix factorization techniques, separate consumer segments no longer need to be extracted in advance and personalized price elasticity can be obtained from the model directly.

By applying the proposed framework, we can provide economic insights from the results in our experiments. These insights include: (1) price does not significantly affect category purchase decisions, suggesting that if the general category of interest is not known, then 'deal' based promotions will be ineffective; (2) price is an important factor in the product choice stage while there is wide variance of price elasticities across categories, products and consumers, which indicates that if the category of a consumer's interest is known, it is effective to target appropriate products and consumers in order to improve the fruitfulness of promotions.

3.2.1 Related Work

Preference matching has been richly studied in the area of recommender systems, where two kinds of approaches of interest have been developed: (1) content-based approaches [105, 129], where explicit user profiles or item information are used as features, and (2) collaborative filtering approaches where preference predictions mainly rely on users' previous behavior [90, 141]. By combining multiple techinques, hybrid recommender systems can be developed to handle a variety of complex scenarios [25, 59]. Matrix factorization techniques have been widely applied for recommender systems due to their accuracy and scalability [16, 50, 90, 145]. Of particular interest, *feature-based* matrix factorization techniques have been proposed [6, 9, 121, 125, 132] and efficient tools (e.g. **SVDfeature**, **libFM**) have been developed [31, 138]. Such ideas have been included in a recently proposed generalized linear mixed model (**GLMix**) [175], which has been deployed in the LinkedIn job recommender system with a scalable parallel block-wise coordinate descent algorithm. We build upon **GLMix** and adapt it to fit different prediction settings, such as

multi-class classification.

Demand systems and price sensitivity have been an ongoing focus of economists [5,39,40]. Three-stage purchase decision decomposition (i.e., category purchase; product choice; purchase quantity), such as we consider here, has been explored in several studies [11, 33, 60, 172, 173]. Customized promotion techniques have been recently proposed for offline and online shopping behavior [37, 172, 173] where individual purchase behavior is considered and optimal promotions are derived. However, these are not completely personalized demand systems and consumer segmentation is required beforehand. In addition, none of these models is considered in the context of large-scale predictive systems.

The idea of price sensitivity in recommender systems for e-commerce has been mentioned as a potential direction in a classic survey [142], though surprisingly we find that this factor has received relatively little attention. Optimization of online promotions in the context of recommendation has been recently studied [77, 78], where the reservation price (i.e., the highest price a customer is willing to pay) is assumed as known information and a complete behavioral model is missing. The most related work is perhaps the price-sensitive recommender system developed in [152]. In their study, however, price is discretized into different levels rather than evaluated numerically and personalization is not thoroughly explored. Such a system thus struggles when quantitatively estimating personalized price sensitivities and cannot effectively support customized promotion strategies.

3.2.2 Methodology

As discussed, we assume that purchase decisions can be predicted in three stages: category purchase incidence, product choice, and purchase quantity. In this section, we first introduce a generalized feature-based matrix factorization (**FMF**) approach, which can be adjusted and applied in different purchase prediction stages. Then we propose a nested framework (**NFMF**) to holistically model the interdependence of these three stages, adopting the above **FMF** model as a

building block in each stage.

A Unified Feature-Based Matrix Factorization Model

We extend the **GLMix** approach [175] and consider a generalized Feature-Based Matrix Factorization model:

$$
\begin{aligned}
& l_{u,i}(t) = link(y_{u,i}(t)) \approx \langle \boldsymbol{\gamma}_u(t), \boldsymbol{\gamma}_i(t) \rangle \\
& = \underbrace{\Big\langle \boldsymbol{w}, \overbrace{\tilde{\boldsymbol{g}}_{u,i}(t)}^{\text{global features}} \Big\rangle}_{\text{global effect}} + \underbrace{\Big\langle \boldsymbol{\gamma}_u^{(o)}, \overbrace{\tilde{\boldsymbol{\gamma}}_i^{(o)}(t)}^{\text{item features}} \Big\rangle + \Big\langle \overbrace{\tilde{\boldsymbol{\gamma}}_u^{(o)}(t)}^{\text{user feature}}, \boldsymbol{\gamma}_i^{(o)} \Big\rangle}_{\text{observed item/user-specific effect}} + \underbrace{\Big\langle \boldsymbol{\gamma}_u^{(l)}, \boldsymbol{\gamma}_i^{(l)} \Big\rangle}_{\text{latent item-user interaction}},
\end{aligned}
\tag{3.1}
$$

where $y_{u,i}(t)$ indicates the label for the interaction of a user u on an item i at timestamp t. and $\langle \cdot, \cdot \rangle$ indicates the inner product. $y_{u,i}(t)$ could be a binary label when predicting category purchase or product choice, or a numeric label when predicting purchase quantity. By applying a link function $link(\cdot)$ (e.g. the logit function, or logarithm function), we can transform the original label matrix into a numeric matrix $\{l_{u,i}(t)\}$ and decompose $l_{u,i}(t)$ as an inner product of $\boldsymbol{\gamma}_i(t)$ and $\boldsymbol{\gamma}_u(t)$. Here $\boldsymbol{\gamma}_i(t)$ and $\boldsymbol{\gamma}_u(t)$ capture both explicit features and latent factors from items and users. We further decompose each prediction into three components: global effects, observed item/user-specific effects and latent item-user interactions.

- **Global effects.** Here $\tilde{\boldsymbol{g}}_{u,i}(t)$ includes a set of provided features for (u,i,t) and $\boldsymbol{w}$ includes a set of global coefficients which need to be estimated and should be consistent for $\forall(u,i,t)$. Such features may include general temporal and spatial factors, such as day-of-week and store location.
- **Observed item/user-specific effects.** The next term can be regarded as an analogy of the random coefficient model [104, 114, 152, 175], which involves explicit features whose coefficients are item- or user-dependent. Here $\tilde{\boldsymbol{\gamma}}_i^{(o)}(t)$ and $\tilde{\boldsymbol{\gamma}}_u^{(o)}(t)$ are explicit item- and user-related features (such as item information, user demographics) while $\boldsymbol{\gamma}_u^{(o)}$ and $\boldsymbol{\gamma}_i^{(o)}$ are

(latent) item- and user-dependent coefficients.

- **Latent item-user interactions.** The last component is designed to capture the remaining latent effects in terms of low-rank user and item factors, where both $\boldsymbol{\gamma}_i^{(l)}, \boldsymbol{\gamma}_u^{(l)}$ are latent parameters that need to be estimated.

Note that considering the identity link function $link(x) = x$, and discarding explicit features and timestamps, the above formulation extends typical matrix factorization formulations:

$$y_{u,i} = b_0 + b_i + b_u + \left\langle \boldsymbol{\gamma}_i^{(l)}, \boldsymbol{\gamma}_u^{(l)} \right\rangle. \tag{3.2}$$

A Nested Factorization Framework

We notice that in different categories, consumers' purchase patterns are different, which requires us to establish a distinct behavioral model for each category. Given a product i in category c, a consumer u, and a timestamp t, suppose we have the following definitions:

$$C_u(t) : \text{consumer } u \text{ selects the category } c \text{ at time } t;$$
$$B_{u,i}(t) : \text{consumer } u \text{ purchases product } i \text{ at } t;$$
$$Q_{u,i}(t) = q : \text{consumer } u\text{'s purchase quantity of } i \text{ at } t \text{ is } q.$$

Thus if we focus on the category c, a consumer's preferences can be represented by the joint probability of buying a certain quantity of a particular product in category c, i.e.,

$$\begin{aligned} & P\big(Q_{u,i}(t) = q,\, B_{u,i}(t),\, C_u(t)\big) \\ = & \underbrace{P\big(C_u(t)\big)}_{\substack{\text{category} \\ \text{preference}}} \times \underbrace{P\big(B_{u,i}(t) \mid C_u(t)\big)}_{\substack{\text{conditional} \\ \text{product preference}}} \times \underbrace{P\big(Q_{u,i}(t) = q \mid B_{u,i}(t),\, C_u(t)\big)}_{\text{conditional quantity preference}}. \end{aligned} \tag{3.3}$$

This joint probability can be regarded as a product of three conditional probabilities which represent the preferences in previous purchase stages. By adopting different link functions in the

previous **FMF** formulation, these three preferences can be estimated by **L**ogistic, **C**ategorical, and **Q**uantity-based **FMF** models.

- **Category Purchase (L-FMF)**. For a given category c, we have the following *logistic* probability

$$\mu_u(t) \coloneqq P_{\Theta_{cate}}\big(C_u(t)\big) = \sigma\big(s_u^{(cate)}(t)\big), \tag{3.4}$$

  where $\sigma(\cdot)$ is the sigmoid function. Here $s_u^{(cate)}(t)$ is a category preference score, factorized using Eq. (3.1), where we have only one general 'item,' i.e., the category c.

- **Product Choice (C-FMF)**. Next we estimate the probability of selecting a product within a category as a multinomial distribution via a softmax formulation:[3]

$$\eta_{u,i}(t) \coloneqq P_{\Theta_{prod}}\big(B_{u,i}(t) \mid C_u(t)\big) = \frac{\exp(s_{u,i}^{(prod)}(t))}{\sum_{i'} \exp(s_{i',u}^{(prod)}(t))}. \tag{3.5}$$

  Similarly, we apply Eq. (3.1) to factorize the product preference score $s_{u,i}^{(prod)}(t)$.

- **Purchase Quantity (Q-FMF)**. Purchase *quantity* can be represented as a positive integer in $\{1,2,\ldots\}$ and follows a shifted Poisson distribution:

$$P_{\Theta_{quant}}\big(Q_{u,i}(t) = q \mid B_{u,i}(t),\, C_u(t)\big) = \frac{z_{u,i}(t)^{q-1} \exp(-z_{u,i}(t))}{(q-1)!}, \tag{3.6}$$

  where $z_{u,i}(t) = \exp(s_{u,i}^{(quant)}(t))$. Again we apply Eq. (3.1) to factorize the quantity preference score $s_{u,i}^{(quant)}(t)$. Notice that the conditional expectation of purchase quantity can be calculated as

$$\hat{q}_{u,i}(t) \coloneqq \mathbb{E}_{\Theta_{quant}}\big(Q_{u,i}(t) \mid B_{u,i}(t), C_u(t)\big) = z_{u,i}(t) + 1, \tag{3.7}$$

  which can be regarded as an estimate of $Q_{u,i}(t)$.

[3]Note that given the fine-grained categories in our data (e.g. 'organic milk'), the multinomial assumption can be justified in most cases. If this were badly violated when users purchase several different products in the same category, this formulation is still helpful as providing the preference-based product ranked list is sufficient in the personalized promotion and recommendation scenario.

Finally, we let $\Theta_{cate}, \Theta_{prod}, \Theta_{quant}$ denote the sets of parameters involved in category purchase incidence, product choice and purchase quantity prediction respectively.

Model Inference

Since the three purchase stages are heterogeneous, we assume $\Theta_{cate}, \Theta_{prod}, \Theta_{quant}$ are separate parameter sets. Models for each stage can then be inferred independently. The proposed framework inherits the scalability of matrix factorization techniques, where efficient algorithms such as stochastic gradient descent can be applied [23]. We optimize all terms following the principle of maximum likelihood estimation (**MLE**). For a given category, we have the following likelihood functions for category purchase, product choice and purchase quantity:

$$\begin{aligned}
\mathscr{LL}_{cate} &= \sum_{u,t} \Big(c_u(t) \log \mu_u(t) + (1 - c_u(t)) \log(1 - \mu_u(t)) \Big), \\
\mathscr{LL}_{prod} &= \sum_{u,i,t} b_{u,i}(t) \log \eta_{u,i}(t), \\
\mathscr{LL}_{quant} &= \sum_{u,i,t} \Big((q_{u,i}(t) - 1) \log z_{u,i}(t) - z_{u,i}(t) \Big) + const,
\end{aligned} \tag{3.8}$$

where *const* is a term independent of the parameters Θ_{quant}, $c_u(t)$, $b_{u,i}(t)$ and $q_{u,i}(t)$ are corresponding labels for $C_u(t)$, $B_{u,i}(t)$ and $Q_{u,i}(t)$.[4]

Particularly for product choice, consumer purchase behavior is a kind of implicit feedback, in the sense that *not* purchasing a particular product does not necessarily indicate that a consumer dislikes it. Thus rather than predicting if a product is selected via **MLE**, we can instead optimize a criterion that says purchased products are simply 'more preferred' than non-purchased ones. This type of optimization criterion is captured by Bayesian Personalized Ranking (**BPR**) [139], a

[4] $c_u(t) = 1$ indicates the incidence of $C_u(t)$ and $b_{u,i}(t) = 1$ indicates the incidence of $B_{u,i}(t)$

technique that approximately optimizes the area under the curve in terms of product rankings, i.e.,

$$AUC = \frac{1}{N}\sum_{u,t}\frac{1}{|I^+_{u,t}||I^-_{u,t}|}\sum_{i\in I^+_{u,t},i'\in I^-_{u,t}}\delta(s^{(prod)}_{u,i}(t) > s^{(prod)}_{i',u}(t)), \tag{3.9}$$

where N is the total number of shopping trips for all consumers, $I^+_{u,t}$ is composed of the products selected by consumer u at timestamp t and $I^-_{u,t}$ includes (a random sample of) products which were not selected. Here $\delta(\cdot)$ is an indicator function. $\delta(s^{(l)}_{u,i}(t) > s^{(l)}_{i',u}(t)) = 1$ indicates that the consumer u prefers product i to product i' (at timestamp t). In practice, we maximize following objective function

$$\mathscr{LL}_{BPR} = \sum_{u,i,t} b_{u,i}(t)\sum_{i'\neq i}\log p_{i>i',u}(t) = \sum_{u,i,t} b_{u,i}(t)\sum_{i'\neq i}\log\sigma\big(s^{(prod)}_{u,i}(t) - s^{(prod)}_{i',u}(t)\big). \tag{3.10}$$

When optimizing the parameters above we adopt a simple ℓ_2 regularization procedure in order to avoid overfitting.

Price Elasticity Estimation

We introduce the concept of 'price elasticity' to model the product price sensitivity, which is a popular measure in economics and can be defined as the responsiveness of a product's purchase quantity (or probability) to changes in its price ('self elasticity') or another product's price ('cross elasticity') [34, 60]. Self elasticity values are usually negative. Larger absolute values of elasticity indicate higher price sensitivity, which means if the product price drops, its purchase probability or purchase quantity will increase accordingly. Since products within a category are often the same kind of commodities (and likely to be substitutes), the cross elasticity values in the product choice stage are usually positive, which indicates that if the product price drops, purchase probabilities of other products within the same category will decrease.

Suppose product prices are involved in previous **FMF** models by logarithmic trans-

formations, and $P_i(t)$ is defined as the price of product i at timestamp t. Due to the linear representation of **FMF**, for a product i, we can represent the previous preference scores $s_u^{(cate)}(t), s_{u,i}^{(prod)}(t), s_{u,i}^{(quant)}(t)$ as

$$\begin{aligned} s_u^{(cate)}(t) &= r_u^{(cate)}(t) + l_u^{(cate)} + \sum_i \beta_{u,i}^{(cate)}(t) \log P_i(t), \\ s_{u,i}^{(prod)}(t) &= r_{u,i}^{(prod)}(t) + l_{u,i}^{(prod)} + \beta_{u,i}^{(prod)}(t) \log P_i(t), \\ s_{u,i}^{(quant)}(t) &= r_{u,i}^{(quant)}(t) + l_{u,i}^{(quant)} + \beta_{u,i}^{(quant)}(t) \log P_i(t). \end{aligned} \tag{3.11}$$

where $\beta_{u,i}^{(\cdot)}(t)$ is the coefficient associated with the price of product i, $r_{\cdot,u}^{(\cdot)}(t)$ captures (temporal and spatial) contextual information of the shopping trip (e.g. day-of-week, store location) and $l_{\cdot,u}^{(\cdot)}$ captures consumer u's category loyalty or product loyalty which is independent of the product's price and the environment of the shopping trip.[5] Then we can define the price elasticity of demand in different purchase stages.

- **Category Purchase**. For the probability of category purchase incidence and the price of product i in this category, we can define the elasticity as [6]

$$e_{u,i}^{(cate)}(t) := \frac{d\mu_u(t)}{\mu_u(t)} \Big/ \frac{dP_i(t)}{P_i(t)} \approx (1 - \mu_u(t)) \beta_{u,i}^{(cate)}(t). \tag{3.12}$$

  Based on Eq. (3.12), if we assume that $\beta_{u,i}^{(cate)}(t)$ does not have significant variations and $e_{u,i}^{(cate)}(t) < 0$, the absolute value of $e_{u,i}^{(cate)}(t)$ will decrease as the preference prediction $\mu_u(t)$ increases.

- **Product Choice**. An advantage of our choice-based model is that product competition within a category can easily be modeled. That is, we can model the effect of a product's price change not just to its own purchase probability but other products' purchase probabilities.

[5] Notice that $r_{\cdot,u}^{(\cdot)}(t)$, $l_{\cdot,u}^{(\cdot)}$ and $\beta_{u,i}^{(\cdot)}(t)$ can be composed of both implicit parameters and explicit features.

[6] This equation can be derived based on the fact that $d(\log(x)) \approx dx/x$.

To do so we define the self elasticity of i as

$$e_{u,ii}^{(prod)}(t) := \frac{d\eta_{u,i}(t)}{\eta_{u,i}(t)} \Big/ \frac{dP_i(t)}{P_i(t)} \approx (1-\eta_{u,i}(t))\beta_{u,i}^{(prod)}(t). \tag{3.13}$$

As with Eq. (3.12) if $\beta_{u,i}^{(prod)}(t)$ does not vary significantly and $e_{u,ii}^{(prod)}(t) < 0$, the absolute value of $e_{u,ii}^{(prod)}(t)$ will decrease as the associated preference prediction increases. For two products i and i', we have the cross elasticity (how a price change for i affects the sales of i')

$$e_{u,ii'}^{(prod)}(t) := \frac{d\eta_{i',u}(t)}{\eta_{i',u}(t)} \Big/ \frac{dP_i(t)}{P_i(t)} \approx -\eta_{u,i}(t)\beta_{u,i}^{(prod)}(t). \tag{3.14}$$

Notice that $\eta_{u,i}(t)e_{u,ii}^{(prod)}(t) + \sum_{i'\neq i}\eta_{i',u}(t)e_{u,ii'}^{(prod)}(t) = 0$, which indicates that total choice shares must be conserved at the product selection level regardless of price fluctuations.

- **Purchase Quantity**. If we use the conditional expectation Eq. (3.7) as the estimation of the conditional purchase quantity, we have the following elasticity definition:

$$e_{u,i}^{(quant)}(t) := \frac{d\hat{q}_{u,i}(t)}{\hat{q}_{u,i}(t)} \Big/ \frac{dP_i(t)}{P_i(t)} \approx (1-\frac{1}{\hat{q}_{u,i}(t)})\beta_{u,i}^{(quant)}(t). \tag{3.15}$$

In this scenario, if the variance of $\beta_{u,i}^{(quant)}(t)$ is limited and $e_{u,i}^{(quant)}(t) < 0$, the absolute value of price elasticity will increase as consumers' preferences increase.

Notice that an advantage of the nested **FMF** framework is that these three elasticities are additive. If we consider the price elasticity for the whole shopping trip, since

$$\mathbb{E}Q_{u,i}(t) = \mathbb{E}\big(Q_{u,i}(t) \mid B_{u,i}(t),\, C_u(t)\big) \times P\big(B_{u,i}(t) \mid C_u(t)\big) \times P\big(C_u(t)\big) = \hat{q}_{u,i}(t)\eta_{u,i}(t)\mu_u(t)$$

then this elasticity can be decomposed as

$$e_{u,i}^{*}(t) = \frac{d\mathbb{E}Q_{u,i}(t)}{\mathbb{E}Q_{u,i}(t)} \Big/ \frac{dP_i(t)}{P_i(t)} = e_{u,i}^{(cate)}(t) + e_{u,ii}^{(prod)}(t) + e_{u,i}^{(quant)}(t). \tag{3.16}$$

3.2.3 Experiments

We evaluate the proposed nested feature-based matrix factorization framework for consumer preference prediction and price sensitivity estimation on two real-world grocery store transaction datasets. For consumer preferences, we evaluate the proposed **FMF** model's ability to make satisfying purchase predictions in terms of category purchase incidence, product choice and purchase quantity estimation. In addition, we provide analysis of the price elasticity estimations and discuss the economic insights behind these observations.

We consider two real-world datasets of supermarket transactions. ***MSR-Grocery*** is a new dataset of convenience store transactions from a grocery store in the Seattle area; since this dataset is proprietary, we also evaluate our method on the public ***Dunnhumby*** dataset to ensure the reproducibility and extensibility of our results. Note that both datasets contain instances of variability in the price of a given product due to promotions, making them an ideal platform to study the effect of price variability on consumer behavior.

- ***Dunnhumby***. The first dataset is the *The Complete Journey* dataset published by ***Dunnhumby***.[7] This dataset includes transactions over two years from around two thousand households who are frequent shoppers at multiple stores of a retailer. Three-level category information is provided in this dataset: department, commodity description, and sub-commodity description. Here we regard the most specific one as the category indicator. We filter out small stores, infrequent shoppers, rare products, tiny categories, and finally obtain around 531 thousand product transactions[8] from 98 thousand shopping trips by 799 consumers at 108 stores, across 4,247 products and 104 categories. Consumer demographic information (household age, marital status, income, homeowner description, household size, number of children, etc.) and product related information (retailer price, coupon information, manufacturer, brand, size, description, etc.) are also included. We follow the dataset specification and

[7] `https://www.dunnhumby.com/sourcefiles`

[8] Each product transaction is for a specific product in a shopping trip.

Table 3.1: Basic dataset statistics.

	#product transactions	#shopping trips	#users	#trips per user
Dunnhumby	531,201	98,020	799	123
MSR-Grocery	152,021	53,075	1,228	43
	#products	**#stores**	**#categories**	**#products per category**
Dunnhumby	4,247	108	104	42
MSR-Grocery	1,929	1	55	35

calculate the actual product price based on the retailer price and promotion information. By comparing the actual price and retailer price, we find that 62% of the products in transaction logs associated with these frequent shoppers were sold on sale.

- ***MSR-Grocery***. We collected eight months of transactions from a single (anonymous) convenience/grocery store in the Seattle area. After removing invalid transactions, infrequent shoppers, rare products, tiny categories, we keep about 152 thousand product transactions from 53 thousand distinct shopping trips by 1,228 frequent consumers across 1,929 popular products in 55 categories. Some product-related features (actual price, package size, size, description) are included, though we cannot obtain any consumer demographics due to the lack of a loyalty program. Since the complete retailer price history is not available, we regard the maximum price in the transaction logs as the retailer price and compare it with the actual price. Ultimately around 50% of the products were sold on sale in this dataset.

Detailed statistics of above two datasets are included in Table 3.1.

Feature Instantiation

Recall that in the general **FMF** representation in Eq. (3.1), both observed features and latent variables are involved. In this section, we will describe the general philosophy of feature design in the context of the consumer behavior model, and the specific features used in each

purchase stage for each dataset.

- **Category Purchase.** For category purchase prediction, three global features ($\tilde{\boldsymbol{g}}_{u,i}(t)$) are considered: (1) consumer u's previous category purchase frequency, which is used to capture u's category preference; (2) category purchase quantity in u's last shopping trip, which is included to capture u's inventory information; (3) prices of products in the given category c. Since popular products may have more significant effects compared with unpopular products from the same category, we transform product prices into log-scale, weighted by their cumulative sold quantities. Since we have only one general 'item' (i.e., the selected category) at this stage, we only consider a simple consumer bias term and ignore latent item-user interactions.
- **Product Choice.** Similarly for a product i, we include the following global features: (1) previous product purchase frequency by the consumer u; (2) current price of the product i ($\log P_i(t)$). Product biases and consumer biases are included. $\log P_i(t)$ is also considered in the item features ($\tilde{\boldsymbol{\gamma}}_i^{(o)}(t)$) such that each consumer and each product has their own price-related coefficients. Latent item-user interaction can be considered if provided product-related and consumer-related features are not sufficient.
- **Purchase Quantity.** For a product i, we consider the consumer u's previous average purchase quantity of the product and its current price as a global effect.

Besides the above mentioned features, additional feature configurations on the ***Dunnhumby*** dataset and the ***MSR-Grocery*** dataset can be found in Table 3.2.

Price History Recovery

In real cases, the complete product price history may be unavailable. Given the transaction logs, we can only observe the prices of those products sold at a certain timestamp. However, as we claimed in the previous section, prices of unsold products ought to be included in the model as well, which requires us to attempt complete price history recovery. Specifically, we applied

Table 3.2: Specific features applied in **FMF** on the ***Dunnhumby*** and ***MSR-Grocery*** datasets. Notice that coefficients for the item intercept and user intercept indicate consumer bias and product bias respectively.

(a) Category Purchase

Dataset	**global features** ($\tilde{\boldsymbol{g}}_{u,i}(t)$)	**item features** ($\tilde{\boldsymbol{\gamma}}_u^{(o)}(t)$)
Dunnhumby	category purchase freq., last purchase quant., day-of-week, storeID, household demographics, **prices** of all products	intercept
MSR-Grocery	category purchase freq., last purchase quant., day-of-week, **prices** of all products	intercept

(b) Product Choice

Dataset	**global features** ($\tilde{\boldsymbol{g}}_{u,i}(t)$)	**item features** ($\tilde{\boldsymbol{\gamma}}_i^{(o)}(t)$)	**user features** ($\tilde{\boldsymbol{\gamma}}_u^{(o)}(t)$)
Dunnhumby	product purchase freq., product **price**, **price***freq., **price***day-of-week, **price***storeID	intercept, product **price**, product info. (brand, manufacturer, size description)	intercept, product **price**, household demographics
MSR-Grocery	product purchase freq., product **price**, **price***freq., **price***day-of-week	intercept, product **price**, product info. (package size, size description)	intercept, product **price**

(c) Purchase Quantity

Dataset	**global features** ($\tilde{\boldsymbol{g}}_{u,i}(t)$)
Dunnhumby	avg. purchase quant., day-of-week, storeID, product info., household demo., product **price**, **price***(avg. quant.), **price***day-of-week, **price***storeID, **price***(product info.), **price***(household demo.)
MSR-Grocery	avg. purchase quant., day-of-week, product info., product **price**, **price***quantity, **price***day-of-week, **price***product info.

a simple 'hot deck' method [119] for imputing these missing prices, where the transactions are sorted by timestamps and the last observed price of the same product is carried forward to the current missing price. Note that this approach can be implemented efficiently but may generate biased values if people rarely buy products at their original price. Thus we claim that developing stronger approaches to recover the complete price histories could be another important problem which can potentially be explored as future research.

Baselines and Evaluation Methodology

Consumers' previous category purchase frequencies, product purchase frequencies and average purchase quantities can be adopted as three simple baselines – **cateFreq**, **prodFreq** and **avgQuant** for category purchase, product choice and purchase quantity predictions.

We also consider standard logistic regression (**L-Reg**) for category purchase where all the global features in Table 3.2 are included. For product choice, matrix factorization as in Eq. (3.2) (**MF-mle**) is applied to fit the multi-class classification setting.[9] **L-Reg** and **MF-mle** thus yield two learning-based recommendation benchmarks for category purchase and product choice.

Finally, we apply two sets of **FMF**-based methods for all of these three prediction stages: (1) **L-FMF-b**, **C-FMF-b-mle** and **Q-FMF-b** are three **FMF** baselines where all features in Table 3.2 except for product prices are included and the **MLE** optimization criterion is applied; (2) **L-FMF-p**, **C-FMF-p-mle** and **Q-FMF-p** are three full **FMF** models where product prices are added back. Comparing these two sets of baselines, the importance of product prices can be evaluated. In addition to **MLE**, we adopt another method **C-FMF-p-bpr** for product choice where the **BPR** criterion Eq. (3.10) is used to optimize the personalized product ranking (i.e., the *AUC*) directly.

Note that the number of purchase incidences for each category is usually much smaller than the total of those for the remaining categories in the complete transaction logs. Therefore we

[9]The dimension of $\gamma_i^{(l)}, \gamma_u^{(l)}$ is set to 5.

apply the area under the curve (AUC) metric to evaluate the performance of category purchase prediction, which is suited to imbalanced binary prediction tasks [49]. For product choice, in real-world recommender systems, one is often interested in providing satisfactory ranked lists instead of simply predicting incidence. Thus we directly adopt the AUC defined in Eq. (3.9), which measures if the selected product is preferred to those products that were not selected in each shopping trip. Since purchase quantity estimation is a numeric prediction task, we apply the mean absolute error (MAE) to evaluate performance. One advantage of this measure is that the MAE is more robust to outliers than the root mean squared error (RMSE).

Preference Prediction

We chronologically partition shopping trips into 70/10/20 training/validation/test splits. Because of the number of item- and user-related parameters is very large, we set two different coefficients on the ℓ_2 regularizers of the global parameters (λ_1) and the item-/user-related parameters (λ_2). These coefficients are selected on the validation set.[10] All results in this section are reported on the test data.

We evaluate the performance for preference prediction based on the measures described in the previous section.

- **Category Purchase.** Results of category purchase prediction in terms of the AUC for binary classification are shown in Table 3.3a. Compared with the baseline **cateFreq**, category prediction can be significantly improved by incorporating additional features and consumer biases. However, we notice that price has little impact on performance, indicating that it may be difficult to drive consumers' category purchase decisions by altering product prices.
- **Product Choice.** For product choice prediction, we evaluate the product-ranking AUC. Results across different categories are provided in Table 3.3b. Compared with **prodFreq**

[10] λ_1 is selected from $\{0.1, 0.5, 1, 5\}$ and λ_2 is selected from $\{1, 5, 10, 50\}$.

Table 3.3: Mean and standard error of preference prediction results across different categories in different purchase stages on the ***Dunnhumby*** and ***MSR-Grocery*** datasets.

(a) Category purchase prediction (AUC).

Dataset	***Dunnhumby***		***MSR-Grocery***	
	mean	**s.e.**	**mean**	**s.e.**
cateFreq	0.661	0.006	0.643	0.009
L-Reg	0.722	0.006	0.657	0.009
L-FMF-b	0.782	0.005	*0.747*	0.008
L-FMF-p	*0.783*	0.005	0.746	0.007

(b) Product choice (AUC).

Dataset	***Dunnhumby***		***MSR-Grocery***	
	mean	**s.e.**	**mean**	**s.e.**
prodFreq	0.726	0.006	0.727	0.008
MF-mle	0.723	0.006	0.641	0.006
C-FMF-b-mle	0.824	0.005	0.802	0.007
C-FMF-p-mle	0.830	0.005	0.802	0.007
C-FMF-p-bpr	*0.832*	0.005	*0.808*	0.007

(c) Purchase quantity prediction (MAE).

Dataset	***Dunnhumby***		***MSR-Grocery***	
	mean	**s.e.**	**mean**	**s.e.**
avgQuant	0.706	0.033	0.386	0.021
Q-FMF-b	*0.372*	0.023	0.123	0.022
Q-FMF-p	*0.372*	0.025	*0.115*	0.021

Table 3.4: Summary of self price elasticity estimation.

	Dunnhumby			*MSR-Grocery*		
Dataset	**median**	**mean**	**s.d.**	**median**	**mean**	**s.d.**
cate. purchase	-0.001	-0.012	0.029	-0.025	-0.055	0.062
product choice	-0.798	-0.842	0.683	-0.117	-0.242	0.551
purchase quant.	-0.141	-0.196	0.213	-0.004	-0.024	0.067

and **MF-mle**, performance can be improved by incorporating more features and latent factors. We particularly notice the significance of the price feature on the ***Dunnhumby*** dataset by comparing the performance of **C-FMF-b-mle** and **C-FMF-p-mle**. Also in general, **C-FMF-p-bpr** reliably outperforms other **MLE**-based methods by directly optimizing ranking scores.

- **Purchase Quantity.** We include results for purchase quantity prediction in Table 3.3c. Again, performance can be improved by including additional features in the **Q-FMF** model but product price features do not help substantially.

Price Elasticity Estimation

Next we consider price elasticity estimation. Table 3.4 shows summary results (median, mean and standard deviation) of the elasticity distribution across all shopping trips in each purchase stage. Elasticity for product choice is calculated from **C-FMF-p-bpr**.

Based on the results in Table 3.4, price elasticity for category purchase prediction is limited. This indicates that it is hard to drive consumers' desire to purchase items from a particular category by a single product promotion (at least for grocery shopping). Compared with category and quantity prediction, product choice is the most price sensitive stage (in terms of elasticity) in the decision making process, while price still serves as an important, but less significant, feature for quantity prediction (especially on the ***Dunnhumby*** dataset). We also notice that consumers in the ***Dunnhumby*** dataset are more price-sensitive than those in the

Table 3.5: The eight most price sensitive categories regarding three different purchase stages on the ***Dunnhumby*** and ***MSR-Grocery*** datasets. Values are the median self-elasticities within each category. Categories marked with * are composed of featured products at special locations of the store.

(a) Category Purchase

Dunnhumby		***MSR-Grocery***	
bacon (economy)	-0.04	broth	-0.19
soft drinks (20/24pk)	-0.04	spices	-0.17
beef (lean)	-0.01	popcorn	-0.15
garbage compactor	-0.01	energy drinks	-0.15
hot dogs	-0.01	chocolate	-0.14
pork rolls	-0.01	pizza	-0.11
salad	-0.01	tortilla chips*	-0.10
baby diapers	-0.01	protein bars	-0.10

(b) Product choice

Dunnhumby		***MSR-Grocery***	
bacon (economy)	-2.59	eggs	-1.65
milk (white)	-1.96	coffee	-1.18
butter	-1.84	chips	-1.07
cereal (family)	-1.79	bottle water	-1.06
juice	-1.78	tortilla chips*	-0.98
tuna	-1.77	bottled dill	-0.82
cereal (kids)	-1.67	fresh bread	-0.80
pork rolls	-1.67	candy	-0.80

(c) Purchase Quantity

Dunnhumby		***MSR-Grocery***	
mac & cheese	-0.47	chocolate	-0.12
soft drink (12/15/18pk)	-0.44	candy	-0.06
bacon (economy)	-0.41	energy drinks	-0.05
hot dogs	-0.41	mac & cheese	-0.04
milk (white)	-0.34	sausage	-0.04
pork rolls	-0.33	frozen fruit	-0.04
frozen dinner	-0.32	spices	-0.04
facial tissue	-0.32	soft drinks	-0.04

MSR-Grocery dataset in the product choice and purchase quantity stages. One possible reason is that the ***MSR-Grocery*** dataset is collected from a convenience store, where people usually have certain targets in mind and are less likely to seek a large inventory of products. On the other hand, the ***Dunnhumby*** dataset is composed of household-level shopping transactions where consumers may be more likely to redeem promotions and purchase more products.

In Table 3.5, we provide details of the eight most price-sensitive categories in each purchase stage. From Table 3.5b, we notice that consumers tend to select the most inexpensive products when shopping for meat (bacon, pork rolls, tuna), eggs, drinks (water, juice, coffee), cereal and snacks (potato chips, candy). In addition, from Table 3.5c we observe that consumers are more likely to stock products which have relatively long shelf lives (e.g. frozen food, soft drinks) if appropriate promotions are offered. Some featured categories (categories with promotions and located in designated areas) in the ***MSR-Grocery*** dataset appear in Table 3.5a and Table 3.5b, which indicates that a combination of promotions and advertisements may help to affect consumers' purchase decisions.

3.2.4 Case Study: Bacon

Besides showing the overall preference prediction performance and the price elasticity distribution across the 104 categories in the ***Dunnhumby*** and 55 categories in the ***MSR-Grocery*** dataset, we provide detailed explorations of the most price sensitive category from the ***Dunnhumby*** dataset in the product choice stage: 'bacon (economy).' A summary of product prices in this category is included in Table 3.6, where price variabilities can be observed for all products except *product 10*. We also include the total quantity sold for each product in Table 3.6, where we notice that products with moderate prices are more popular than others.

Table 3.6: Summary of product prices and sold quantities on ‘bacon (economy)’.

Product	prod. 1	prod. 2	prod. 3	prod. 4	prod. 5	prod. 6	prod. 7	prod. 8	prod. 9	prod. 10	prod. 11
mean	$1.95	$2.99	$2.88	$3.09	$3.02	$3.08	$3.38	$3.83	$3.90	$5.99	$5.66
standard deviation	$0.14	$0.72	$0.62	$0.67	$0.71	$0.77	$0.86	$0.66	$0.54	$0.00	$0.53
minimum	$1.49	$2.00	$2.00	$2.00	$1.50	$2.50	$2.50	$0.99	$2.49	$5.99	$3.99
maximum	$1.99	$3.99	$3.99	$3.99	$3.99	$4.39	$4.39	$4.49	$4.99	$5.99	$5.99
# unique values	4	4	4	4	8	5	5	4	5	1	3
quantity sold	204	373	215	455	730	507	173	64	65	32	30

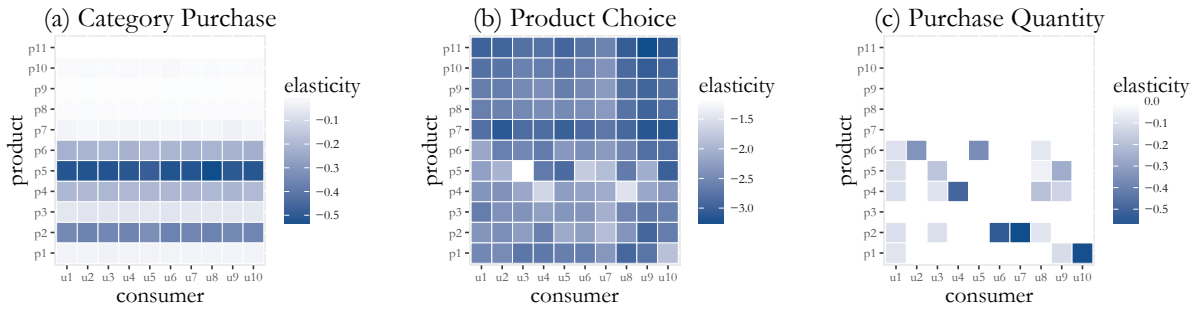


Figure 3.2: Heatmaps of consumer-specific price elasticity in different purchase stages for the example category 'bacon (economy).' Darker blocks indicate higher price sensitivity.

Preference vs. Representative Features

We find that the estimated coefficient on 'bacon (economy)' for *category frequency* in the category purchase stage is 0.28, which indicates that a consumer's previous category purchase frequency is still positively related to the category preference. Also the estimated coefficient for *last purchase quantity* is -0.22, which means if consumers purchased a substantial volume of economy bacon products in their previous shopping trips, they may avoid making the same category purchase in their current shopping trip. For the product choice stage and the purchase quantity stage, we find that the estimated coefficients for *product frequency* and *average purchase quantity* are 0.28 and 0.20, which indicates these two features are positively correlated with preferences as well.

Personalized Product-Specific Price Sensitivity

Among the 11 products in the example category 'bacon (economy)' there are 448 consumers who have purchased products in this category. We randomly select 10 consumers and calculate their average price elasticity for each 'bacon (economy)' product in terms of category purchase, product choice and purchase quantity decisions. Heatmaps of the results are shown in Figure 3.2. We notice that within the 'bacon (economy)' category, different consumers and products may have significantly different price sensitivities in each of the three stages, though the

personalized elasticity is not obvious in Figure 3.2a since the user-specific price coefficient is not considered in the first stage. By setting appropriate thresholds for price elasticity, we can easily uncover those price sensitive consumer-product pairs in Figure 3.2 and customize promotion strategies accordingly. In addition, we find that in Figure 3.2a, consumers are more sensitive to the prices of *products 2–6* in the category purchase stage, which indeed are popular products as we observed in Table 3.6. This implies that while it is hard to increase the possibility of category purchase incidence, promotions on popular products will be more effective than others in terms of category purchase.

Preference vs. Price Sensitivity

Recall we claimed that if the variances of price-associated coefficients in Eq. (3.12), Eq. (3.13), Eq. (3.15) are limited, then consumers with high preference scores will be relatively insensitive to price changes as far as category and product choice is concerned, but they tend to be price sensitive with respect to purchase quantity. Again taking 'bacon (economy)' as an example, in Figure 3.3 we show the relationship between preferences and price sensitivities in different purchase stages. We notice that all elasticity values are negative, which is consistent with the intuition that purchase probability will increase if product price drops. Here absolute price elasticity values are generally negatively correlated with preferences in category purchase and product choice, but positively correlated with purchase quantity, which indeed verifies our previous arguments about the relationship between preference and price sensitivity. In Figure 3.3b, we notice that 'low-preference' consumers have larger variations in price sensitivity than 'high-preference' consumers. This is possibly because high-preference consumers' preferences dominate purchase decisions (i.e., $1 - \eta_{u,i}(t)$ is close to zero in Eq. (3.13)) and they tend to purchase a product no matter its price. On the other hand, if a product is not preferred by a consumer, this could be either because the price is too high to trigger a purchase, or because the consumer simply dislikes the product. In Figure 3.3c, we observe that those consumers with strong preferences are not the

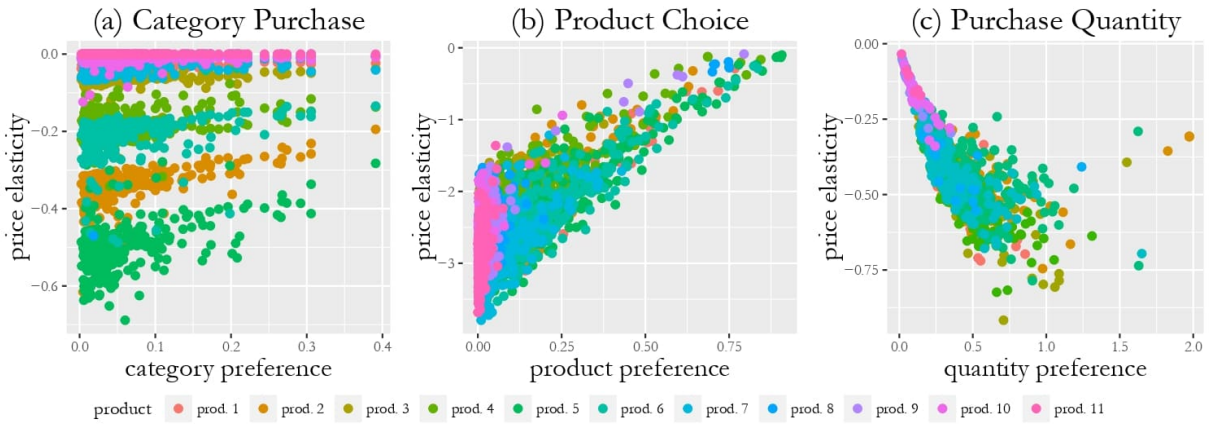


Figure 3.3: Scatter plots between preference prediction and price elasticity estimation in different purchase stages for the example category 'bacon (economy)'. Note that axes within each subfigure are scaled based on their own ranges.

most price-sensitive consumers. This observation is consistent with the intuition that aggressive buyers are more likely to exhaust the potential of purchase quantity due to budget limits so that it would be difficult to increase their purchase quantities by adjusting price.

3.3 Product Complementarity and Compatibility

Many modern and traditional recommendation techniques depend on learning latent representations of items from interaction data. A traditional example is a latent factor model [90], where a user-item interaction matrix is factorized by low-dimensional user and item terms. These methods in general attempt to recover the original interaction information globally, but may fail to capture subtle and fine-grained semantics of items. Inspired by word embedding techniques proposed for Natural Language Processing (NLP) tasks [115, 116, 130], recent item representation techniques have been developed for e-commerce [17, 56, 149, 153, 158]. In general these approaches are designed to learn representations which can effectively recover product co-occurrences either within a basket or across baskets from the same user. Foregoing hand-crafted feature design, these methods automatically uncover useful (in terms of recommendation)

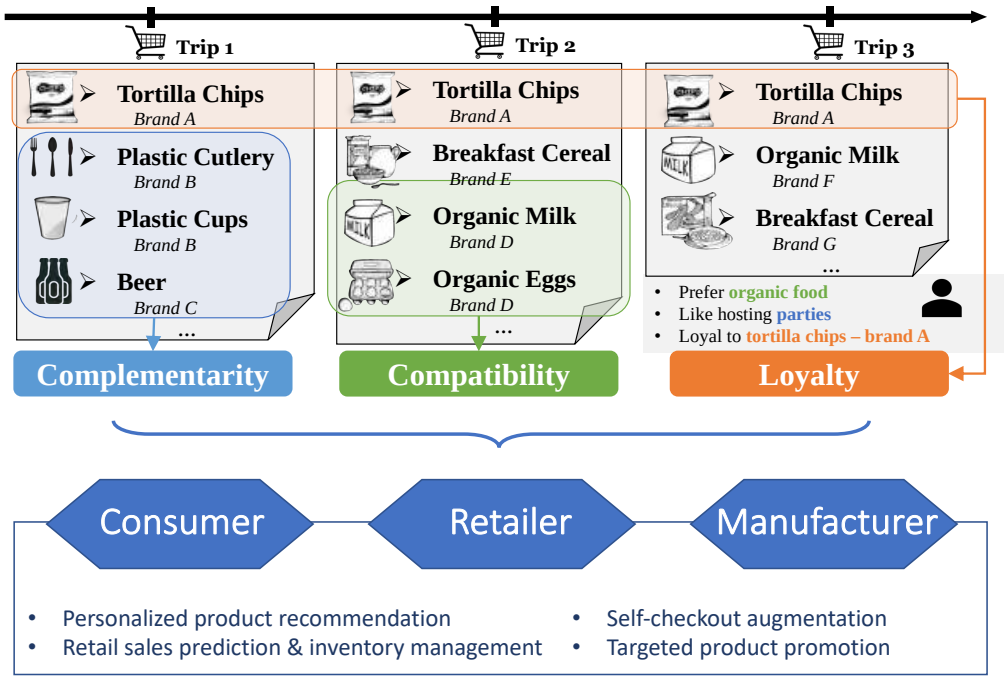


Figure 3.4: Three significant patterns observed in users' grocery baskets (item-to-item *complementarity*, user-to-item *compatibility*, and product *loyalty*) and their applications in the grocery industry.

representations of products. Both modern and traditional techniques have seen wide adoption in real-world e-commerce applications. We hope to examine and adapt the above techniques to shopping transaction data and observe the following patterns.

- **Complementarity.** Users purchase multiple related products in the same basket to fulfill specific needs. These products complement each other in terms of functionality. For example, shoppers may buy plastic cups/forks/knives/plates and beer together for a party (Figure 3.4). Such item-to-item complementarity is critical as it not only captures products' latent functions but also reveals a user's intent in each basket.
- **Compatibility.** As with most e-commerce categories, compatibility between users' preferences and products' properties is paramount in grocery shopping. In addition, we notice that the above latent functions for complementarity may need to match users' preferences as well (e.g. plastic cups and cutlery are more likely to appear in a party lover's basket). This

kind of cohesion inspires us to consider item-to-item complementarity and user-to-item compatibility holistically in our representation learning model for grocery shopping.

Our primary goal in this study is to leverage the above properties in the grocery shopping domain, and to develop a framework to understand the semantics of users' purchases. The representations we learn are generalizable and support tasks like automatic product categorization and personalized recommendation for grocery shopping at scale.

Summary of Contributions

Inspired by the confluence of complementarity and compatibility, we focus on the core component in grocery transaction data—$(item, item, user)$ triples linked by the same basket, i.e., two items purchased in the same basket from a user, and propose a representation learning model **triple2vec** to recover the above complementarity and compatibility holistically.

We conduct extensive experiments on two public and two proprietary datasets, which cover various grocery store types including conventional physical supermarkets, a convenience store, and an online grocery shopping platform.

Based on the quantitative results from experiments, we demonstrate that the proposed **triple2vec** model is able to generate meaningful (in terms of product classification tasks) and useful (in terms of recommendation tasks) product representations.

3.3.1 Related Work

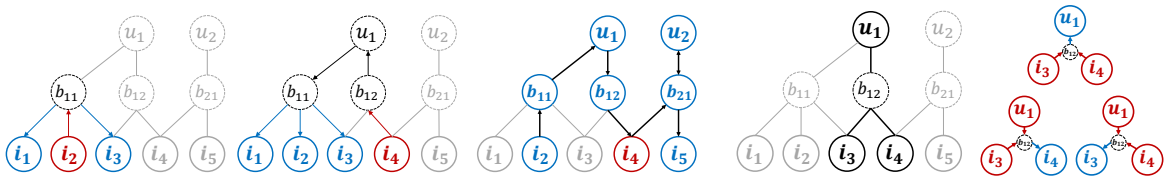
Traditional *model-based* item recommendation methods typically rely on Matrix Factorization (MF) techniques, e.g. via a latent factor model [90]. Of most relevance to grocery shopping are variants of MF for implicit feedback data where only positive signals (e.g. purchases) can be observed [72, 139]. MF-based methods have been extended to sequential recommendation (i.e., predicting items in a shopper's next basket, based on the context of their previous basket) by appropriately unifying MF and Markov Chains [140, 159]. More recently, by considering

both user-to-item interactions and multiple associations among items simultaneously, such factorization techniques have been extended to within-basket recommendation (i.e., recommending products to be added to the current basket) [96]. In general, these models are optimized to directly favor global recommendation metrics which, while effective for recommendation, may fail to capture detailed semantics of products.

Recently, 'neural' representation learning methods including **word2vec** [115, 116] and **GloVe** [130] have achieved success on various NLP tasks. Particularly, the **skip-gram** technique [115, 116] has been widely extended to other domains including e-commerce [17, 56, 149, 153]. For example, **item2vec** was proposed by directly applying the **skip-gram** framework on itemsets, so that it can represent associations among products within the same itemset [17]. **prod2vec** and **bagged-prod2vec** were proposed to learn distributed product representations to support ad recommendations in *Yahoo! Mail* [56], where **skip-grams** are applied to recover product co-occurrence information across the same user's transactions. In order to overcome cold-start problems, several representation learning architectures have been developed to learn product embeddings by incorporating rich meta-data from different sources (e.g. product categories, images, description text) [149, 153, 176]. Moreover, a random-walk based network embedding method **metapath2vec** [44] was proposed to learn node representations from heterogeneous networks; as the relationships among users, baskets, and products can be easily represented as a heterogeneous graph, we consider this as a potential technique which can be applied on grocery shopping transaction data. Details of some representative product embedding learning methods will be provided in Section 3.3.2.

3.3.2 Methodology

We briefly introduce the basic skip-gram-based framework and several representative instantiations. Then we present the proposed representation learning method **triple2vec**, and show how to build downstream product classification and recommendation systems.



(a) item2vec (b) prod2vec (c) metapath2vec (d) triple2vec

Figure 3.5: An illustrative example of different representation learning models. Here $\{u.\}$, $\{i.\}$, $\{b.\}$ are used to represent different users, items, and baskets. In each model, the given node is highlighted in red and the nodes for prediction are highlighted in blue.

Background

Several product representation learning methods are based on the skip-gram framework [116]. Essentially, they seek to find item representations which are useful for predicting contextual (related) items or users, by defining different 'context windows.' In this section, we introduce them as different instantiations of a unified skip-gram framework on a heterogeneous graph (Figure 3.5), whose nodes are composed of different products, users, and baskets. Here we have two different types of links: (1) item-to-basket, indicating that an item is included in a basket, and (2) user-to-basket, indicating that a basket is purchased by a user. On this graph, several existing representation learning objectives can be cast as learning node representations which maximize the (log) likelihood of using a target node v to predict the contextual nodes $\mathscr{C}_v$, i.e.,

$$\mathscr{L}_{sgn} = \sum_{v} \sum_{v' \in \mathscr{C}_v} \log P(v'|v). \tag{3.17}$$

$P(v'|v)$ is commonly defined as $P(v'|v) = \frac{\exp(\boldsymbol{f}_v^T \boldsymbol{g}_{v'})}{\sum_{v''} \exp(\boldsymbol{f}_v^T \boldsymbol{g}_{v''})}$, where $\boldsymbol{f}_v$ and $\boldsymbol{g}_v$ are K dimensional 'input' and 'output' vector representations of a node. We briefly introduce three representative methods of this type as follows:

- **item2vec.** Basket-level skip-grams can be directly applied on this graph, where we treat a particular item as a target node, and the rest of the products in the same basket as contextual nodes [17]. This definition relies on the assumption that products purchased in the same

basket share similar semantics, which intuitively supports within-basket/"bundle" product recommendations. However, such co-purchase relationships may not be sufficient to capture personalized preferences toward products.

- **prod2vec.** Rather than directly applying Eq. (3.17) on baskets, in **prod2vec**, given a target product, the contextual nodes are defined as the products in recent baskets purchased by the same user [56].[11] Unlike the previous method which focuses on within-basket co-purchase relationships regardless of users, this approach emphasizes cross-basket item-to-item relationships for each user.
- **metapath2vec.** As mentioned, transaction logs can be transformed into a heterogeneous network. Therefore, a state-of-the-art network embedding learning method such as **metapath2vec** [44] can be applied here. In this scenario, we need to define a symmetric meta-path scheme: $item \rightarrow basket \rightarrow user \rightarrow basket \rightarrow item$, and generate different random walkers based on this pre-defined scheme. Specifically, we start with a random product, and sample a series of nodes to compose a random walker where each of the nodes consecutively links to the previous one on this meta-path. Then we select a given node, and define its surrounding nodes along the walk as 'contexts' $\mathscr{C}_v$ in Eq. (3.17). A concrete example is included in Figure 3.5c; here we sample a random walker $(i_2, b_{11}, u_1, b_{12}, i_4, b_{21}, u_2, b_{21}, i_5, ...)$, highlight a randomly selected target node i_4 in red and its surrounding nodes in blue.[12] Note that this $item \rightarrow basket \rightarrow user \rightarrow basket \rightarrow item$ meta-path in general captures the semantics of products purchased by the same user but does not reflect product co-purchase relationships explicitly.

[11] We consider the bagged version of **prod2vec** here, where all products purchased in the same contextual basket need to be included together in the objective function.

[12] The neighborhood size is set to 8 in this figure.

Triple2vec: Representations from Triples

Unlike existing skip-gram-based product representations, we focus on the cohesion of each $(item, item, user)$ reflecting two items purchased by the same user in the same basket. Specifically, the transaction logs for training consist of a series of such triples:

$$\mathscr{T} = \{(i, j, u) | i \in I_b \wedge j \in I_b \wedge i \neq j \wedge b \in \mathscr{B}_u \wedge u \in U\}. \tag{3.18}$$

Then we define the cohesion score of each (i, j, u) triple as

$$s_{i,j,u} = \overbrace{\boldsymbol{f}_i^T \boldsymbol{g}_j}^{\text{item-to-item complementarity}} + \underbrace{\boldsymbol{f}_i^T \boldsymbol{h}_u + \boldsymbol{g}_j^T \boldsymbol{h}_u}_{\text{user-to-item compatibility}}, \tag{3.19}$$

where $\boldsymbol{f}_i, \boldsymbol{g}_j$ are two sets of representations for products and $\boldsymbol{h}_u$ represents the embedding vector of a user. The first term in Eq. (3.19) models the complementarity between two products within the same basket, i.e., whether two products exhibit similar semantics in terms of co-occurrence; the second and third terms are used to capture the compatibility between the product and the user, i.e., how well the product's (latent) properties match the user's preferences. A higher cohesion score indicates closer connections among the nodes in the triple. Finally, we aim to learn embeddings which optimize the occurrence likelihood of the training triples $\mathscr{T}$:

$$\mathscr{L} = \sum_{(i,j,u) \in \mathscr{T}} \big(\log P(i|j,u) + \log P(j|i,u) + \log P(u|i,j) \big), \tag{3.20}$$

where $P(i|j,u) = \frac{\exp(s_{i,j,u})}{\sum_{i'} \exp(s_{i',j,u})}$[13] and $P(u|i,j) = \frac{\exp(s_{i,j,u})}{\sum_{u'} \exp(s_{i,j,u'})}$. Essentially for each triple in $\mathscr{T}$, we iteratively 'knock out' a node and use the other two nodes to predict it. Figure 3.5 shows an illustrative example highlighting the difference between the proposed **triple2vec** and skip-gram-based models.

[13] Because of symmetry, by exchanging i and j, $P(j|i,u)$ can be obtained.

Negative Sampling

As with skip-gram-based models, a variation of Noise Contrastive Estimation (NCE) can be applied to approximate the softmax function in Eq. (3.20) and accelerate training [116]. For example, $\log P(i|j,u)$ in Eq. (3.20) can be replaced by

$$\log \sigma(s_{i,j,u}) + \sum \mathbb{E}_{i' \sim P(i)} \log \sigma(-s_{i',j,u}), \tag{3.21}$$

where we sample N negative items from a pre-defined distribution $P(i)$, and $\sigma(x) = \frac{1}{1+\exp(-x)}$. We achieve this through the NCE loss API provided in TensorFlow [3], where $P(i)$ is defined as a log-uniform (Zipf) distribution. Specifically items are sorted in order of decreasing popularity and the probability of each item i being sampled is defined as $P(i) = \log \frac{r_i+2}{r_i+1}$, where r_i denotes the rank of item i. This negative sampling technique is used in all the representation learning methods implemented in our experiments, which empirically accelerates model convergence and improves quantitative performance compared to a uniform sampling strategy.

Representing and Recommending Products

Note that two sets of product embeddings $\boldsymbol{f}_i$ and $\boldsymbol{g}_i$ are learned from **triple2vec**. These embeddings describe the functions and properties of products from different angles, but the inner product between these two captures the cross product relationship—item-to-item complementarity. Therefore, for tasks to evaluate the semantics of products independently (e.g. product classification, competitor search), we follow the protocol in [130] and use the additive composition $\boldsymbol{f}_i + \boldsymbol{g}_i$ as the ultimate representation of each product i, which empirically gives a slight boost in most tasks. However, for predictive tasks where the cross-item relationship needs to be considered (e.g. item-to-item recommendation, complementary product search), we consider the inner product score $\boldsymbol{f}_i^T \boldsymbol{g}_j$ for two items i, j instead.

In particular, we consider two different recommendation scenarios: personalized next-basket product recommendation, and within-basket product recommendation:

- Given a user, when recommending products for the next basket, we replace the product embedding of the given product $\boldsymbol{g}_j$ by the average embedding of all the products in the user's previous baskets. Then we obtain a new preference score $s_{i,u}$ and the purchase probability is estimated as $p_{i,u} = \frac{\exp(s_{i,u})}{\sum_{i'} \exp(s_{i',u})}$.
- If products in the current basket are given, when recommending products to be added in the same basket, we replace $\boldsymbol{g}_j$ by the average embedding of all the products in the given item set.

Note that another set of preference scores could be obtained by exchanging $\boldsymbol{f}_i$ and $\boldsymbol{g}_i$. We also take the average of preference scores generated from these two methods and consider it as a third option. In our experiments, we report results from the method which yields the best validation performance.[14]

3.3.3 Experiments

We evaluate representations learned from **triple2vec** on two public and two proprietary grocery shopping transaction datasets. In order to demonstrate that product embeddings are *meaningful* and *useful*, we evaluate (1) the item classification performance obtained with these representations; and (2) the accuracy of product recommendations obtained by leveraging these representations.

In addition to **triple2vec**, we consider the three methods described in Section 3.3.2: **item2vec** [17], **prod2vec** [56] and **metapath2vec** [44]. For all representation learning methods, we apply the same negative sampling approach, where the number of negative samples in Eq. (3.21) is set to 5.[15] **AdaGrad** [45], a stochastic gradient-based optimization method, is applied to learn all embeddings.

[14] In our experiments, this protocol is applied for all representation learning methods in which two heterogeneous embeddings are involved.

[15] In practice, product bias terms are also added in Eq. (3.17) and Eq. (3.19) for all of these methods to capture overall item popularity.

Table 3.7: Basic dataset statistics.

Dataset	#item	#user	#trans.	#trans./ #user	basket size	#dept.	#dept. ≥ 5	#cat.	#cat. ≥ 5
Dunnhumby	26,780	2,500	269,974	107.99	9.02	31	24	310	255
Instacart	42,987	206,209	3,345,786	16.23	10.10	21	21	134	134
Grocery(WA)	16,497	47,939	360,222	7.51	4.81	34	26	306	177
Grocery(UT)	26,821	60,421	634,733	10.51	9.80	24	22	288	247

Datasets

We consider four real-world grocery transaction datasets, where ***MSR-Grocery (WA)*** and ***MSR-Grocery (UT)*** are two proprietary datasets collected from the Seattle and Salt Lake City areas (respectively). In order to ensure the reproducibility of our results, we also evaluate the performance on two public datasets—***Dunnhumby*** and ***Instacart***.

- ***Dunnhumby.*** *The Complete Journey* dataset from Dunnhumby.[16] Transactions over two years collected from around two thousand households are included in this dataset. Users are frequent shoppers with an average shopping frequency of once per week.
- ***Instacart.*** This dataset was published by *instacart.com* [1], a web service that provides same-day grocery delivery in the US. It contains over 3 million grocery orders from more than 200 thousand users. The specific date of each order is missing but the sequence order of transactions by each user is provided.
- ***MSR-Grocery (WA).*** This dataset is collected from a single convenience store in the Seattle area and was used in Section 3.2. We extend this dataset to include 12 months of transactions from around 360 thousand users. Because of the type of this store, users tend to have fewer transactions and smaller basket sizes compared with other datasets.
- ***MSR-Grocery (UT).*** Finally we collected 8 months of transactions from two mid-size grocery stores in the Salt Lake City area. These two stores are from the same grocery chain

[16] `https://www.dunnhumby.com/sourcefiles`

and include relatively diverse consumers including households and college students.

After removing rare products (fewer than 10 purchases) from these datasets, the basic statistics of these preprocessed datasets are listed in Table 3.7. The following rules are applied to split transaction data into train/validation/test sets: 1) for users who have more than one transaction, their most recent transaction is used for testing; 2) for users who have more than two transactions, their second-to-last transactions is used for validation; 3) all the other transactions are used for training. All embedding learning and recommendation models are learned on the training data, and all hyper-parameters are selected based on validation performance. All recommendation results are reported on the held-out test data.

Product Classification

Hierarchical product categories are provided in all four datasets. We treat the top-level hierarchy as the 'department' and the second-level as the 'category,' and remove small departments and categories with fewer than 5 products (see Table 3.7). We evaluate the quality of the product embeddings learned by different methods over both coarse-grained (department) and fine-grained (category) classification. In particular, we apply a one-vs-all linear logistic regression classifier on the product embeddings, where the hyper-parameter for the l_2 regularizer is selected based on a 5-fold cross-validation.

We fix the embedding dimensionality to $K = 32$ and use half of the products for training and the other half for testing (i.e., label fraction $r = 0.5$). Then we repeat each experiment 10 times, and report the average micro-F1 and macro-F1 scores in Table 3.8.[17] We also vary the embedding dimension K, label fraction r, and report the classification results in Figures 3.6a and 3.6b on the *Dunnhumby* dataset to address the sensitivities of these hyper-parameters [18].

We find that **triple2vec** substantially and consistently outperforms all baselines on both department and category classification. The improvement is increased with larger embedding

[17] All differences are significant at 5% level.

[18] $K \in \{8, 16, 32, 64, 128\}$, $r \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$

Table 3.8: Detailed results on product classification tasks ($K = 32, r = 50\%$, the best performance is underlined). All reported improvements are significant at 1%.

(a) F1 metrics on coarse-grained (department) classification

	Dunnhumby		***Instacart***		***Grocery(WA)***		***Grocery(UT)***	
Method	micro	macro	micro	macro	micro	macro	micro	macro
item2vec	0.665	0.108	0.377	0.283	*0.608*	0.345	0.620	0.239
prod2vec	0.617	0.066	0.330	0.218	0.480	0.212	0.491	0.093
m.2vec	0.627	0.071	0.331	0.221	0.441	0.144	0.484	0.067
triple2vec	*0.669*	*0.114*	*0.382*	*0.294*	0.581	*0.361*	*0.623*	*0.293*

(b) F1 metrics on fine-grained (category) classification

	Dunnhumby		***Instacart***		***Grocery(WA)***		***Grocery(UT)***	
Method	micro	macro	micro	macro	micro	macro	micro	macro
item2vec	0.160	0.046	0.187	0.075	0.518	*0.010*	0.275	0.094
prod2vec	0.087	0.015	0.106	0.030	0.518	0.009	0.119	0.023
m.2vec	0.078	0.007	0.155	0.036	0.518	0.007	0.091	0.008
triple2vec	*0.175*	*0.049*	*0.189*	*0.082*	*0.519*	*0.010*	*0.291*	*0.097*

dimension K. The non-personalized method **item2vec** in general yields better classification results compared with the other two baselines. This may indicate that in order to learn meaningful grocery product representations, within-basket item-to-item complementarity is relatively more significant compared with cross-basket item-to-item relationships or item-to-user relationships. Note that in Table 3.8, the performance of **item2vec** is particularly strong on the convenience store dataset (***MSR-Grocery (WA)***), which may reveal the particular significance of such item-to-item complementarity in this type of stores.

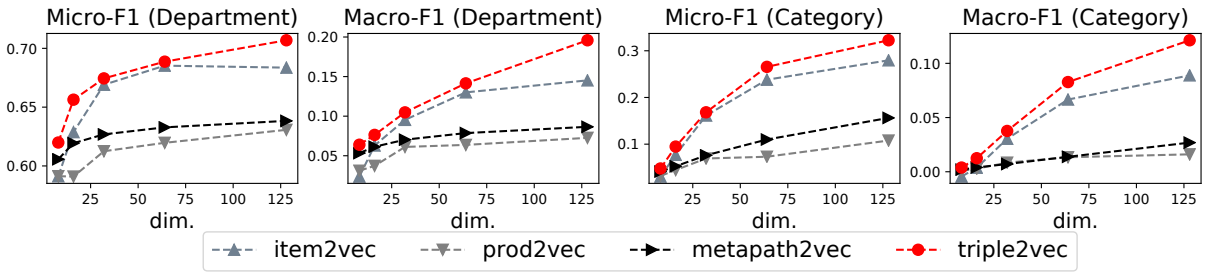
Personalized Recommendation

In addition to product classification tasks, we evaluate the product/user representations on two recommendation tasks: next-basket recommendation and within-basket recommendation. Particularly, we consider the original purchase probability estimated based on embeddings learned

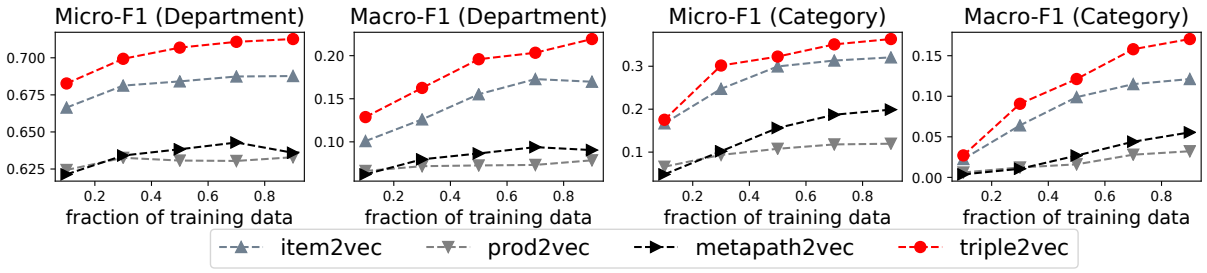
from **item2vec**, **prod2vec** and **metapath2vec** as baselines. For both recommendation tasks, as in [139, 140], we rank products based on the predicted purchase probability, and consider the Area Under the ROC Curve (AUC) as an overall ranking metric, and Normalized Discounted Cumulative Gain (NDCG) as a top-biased evaluation metric. In this section, we report detailed results with a fixed dimension of item/user embeddings $K = 32$ (Table 3.12) and vary it on the *Dunnhumby* dataset for sensitivity analysis (Figure 3.6c).

We first consider recommending products for users' next baskets. The same prediction method in Section 3.3.2 is applied for all representation learning baselines. If a user embedding is not available, we use the average embedding of items in a user's training baskets instead. We further consider two additional baselines: overall item purchase frequency (**itemPop**) and user-wise item purchase frequency (user-wise **itemPop**). As next-basket recommendation is a classic recommendation task, we consider two state-of-the-art *supervised* implicit-feedback recommendation baselines as well: (1) **BPR-MF** [139], an item recommendation model which factorizes the user-item compatibility by approximately optimizing the AUC ranking metric, and (2) **FPMC** [140], where sequential information (via a first-order Markov Chain) is considered in addition to user-to-item compatibility. Detailed results on this task are reported in Table 3.12a. In general **triple2vec** and **metapath2vec** outperform other embedding learning methods, as both explicitly model user-item compatibility during the representation learning process.

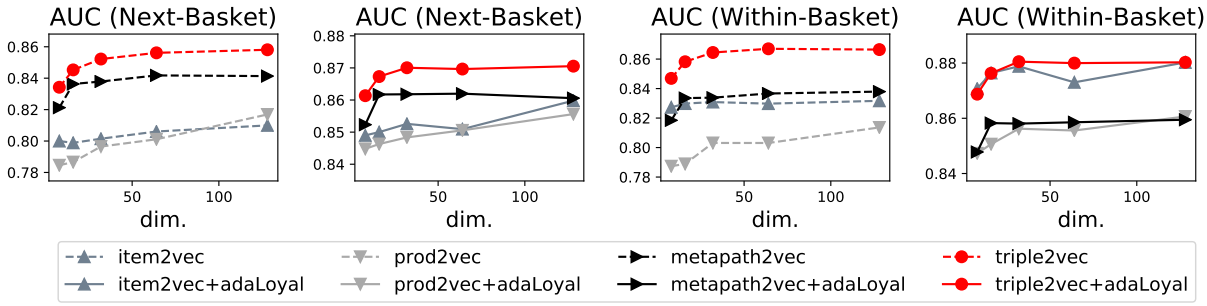
Next we consider a setting where we assume some products in the basket are given and we recommend 'complimentary' products to be added to the basket. Specifically, for a transaction which contains more than one item, we assume half of the products are given and predict the remaining half. For **metapath2vec**, **itemPop** and user-wise **itemPop**, we directly apply next-basket predictions as they do not explicitly account for item-to-item relationships. For **item2vec** and **prod2vec**, we use the item-to-item complementarity score for preference prediction (i.e., $p_{i,uj} \propto \exp(\boldsymbol{f}_i^T \boldsymbol{g}_j)$). Detailed results are included in Table 3.12b and **triple2vec** still dominates other methods in most cases. Besides, we notice that **item2vec** becomes a competitive method in



(a) Dimension K (Department and Category Classification, $r = 0.5$)



(b) Label Fraction r (Department and Category Classification, $K = 128$)



(c) Dimension K (Next-Basket and Within-Basket Recommendation)

Figure 3.6: Sensitivity analysis in product classification and recommendation tasks on the ***Dunnhumby*** dataset.

this scenario and outperforms other baselines. This pattern can be observed when experimenting with different numbers of embedding dimensions as well (see Figure 3.6c). This suggests that by explicitly including within-basket item-to-item interactions in the preference score, item-to-item complementarity can be captured and thus improve within-basket recommendations.

Table 3.9: Complement and competitor search for "Banana" and "Organic Banana" in the *Instacart* dataset. Note the z-normalized complementarity score and the cosine similarity score are shown in the second and last columns.

(a) Product Query: "Banana"

Complements	**Score**	**Competitors**	**Score**
Whole Milk With Vitamin D	3.46	Fuji Apple	0.97
Plain Yogurt	3.11	Honeycrisp Apple	0.96
Apple Blueberry Granola	3.06	Cucumber Kirby	0.93
Orange Navel	3.01	Large Lemon	0.92
Milk Chocolate Nutrition Shake	2.99	Large Grapefruit	0.92

(b) Product Query: "Organic Banana"

Complements	**Score**	**Competitors**	**Score**
Organic Papaya	3.72	Organic Strawberries	0.96
Organic 2% Milk	3.69	Organic Raspberries	0.94
Carbonated Water	3.66	Organic Blueberries	0.94
Organic Bosc Pears	3.61	Organic Hass Avocado	0.93
Organic Applesauce	3.55	Organic Large Extra Fuji Apple	0.92

3.3.4 Case Studies

In addition to our quantitative results, we conduct case studies on our largest public dataset ***Instacart*** to demonstrate the effectiveness of the product embeddings.

Here we demonstrate the effectiveness of our product embeddings by showing how they can be used to search for 'complementary' and 'substitutable' products. We calculate the complementarity score (i.e., $\boldsymbol{f}_i^T \boldsymbol{g}_j$) between all products and the given query product, in order to retrieve the top five complements. For substitutable products, we use the additive composition $\boldsymbol{f}_i + \boldsymbol{g}_i$, and retrieve the top five products (essentially i's competitors) based on cosine similarity. In Table 3.9 we query the two most popular products "Banana" and "Organic Banana." We observe that milk, yogurt, and granola are likely to be complements for bananas while other fresh fruits can be regarded as similar products/competitors. Another interesting pattern is that most retrieved products for "Organic Banana" are organic products while none are for the (non-organic)

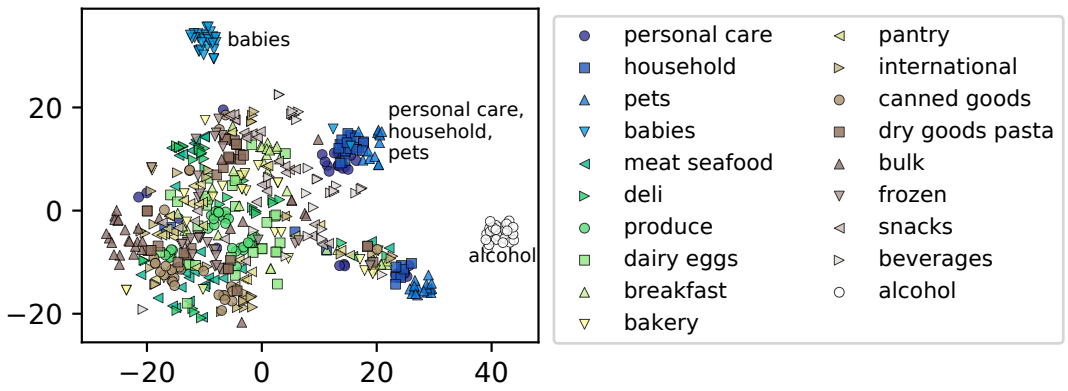


Figure 3.7: 2d t-SNE projections of the 32-dimensional product embeddings learned from **triple2vec** on the *Instacart* dataset.

"Banana." This indicates that our representation learning method **triple2vec** can capture latent properties (e.g. "organic") of products, which might be particularly useful when recommending products to match users' fine-grained preferences.

For each department in *Instacart*, we select the 30 most popular products and visualize the low-dimensional product representations learned from **triple2vec** in Figure 3.7 via t-SNE [106]. In this figure, we notice that **triple2vec** automatically organizes these products around different functions. For example, products in the personal care, household and pets departments are separated from food products (e.g. produce, meat/seafood, dairy/eggs). In addition, two relatively isolated departments – *babies* and *alcohol* can be observed in this dataset. Note these two are relatively loyal but infrequently purchased departments (see Table 3.11b), which may reflect users' unique shopping patterns. Purchases of baby products are normally necessities, which results in relatively tight connections. Similarly, alcohol products are different from other daily food consumptions in nature. Also, purchasing such products usually incurs additional 'costs' to users (e.g. providing valid identification). Therefore, users' shopping patterns in this department are dramatically different from others.

3.4 Product Loyalty

As showed in Figure 3.4, a pronounced pattern in shopping transactions such as grocery shopping baskets is that users tend to exhibit *'loyalty'* towards certain products, i.e., repeatedly purchasing the same product while rarely switching brands among alternatives. Such behavior is often contrary to the assumptions implicit in conventional recommendation models: typically if two products have similar representations and match a user's preferences (or needs), either could be recommended. However, if a user is loyal to one product, then its alternatives have systematically low probability of being purchased. In fact, simple user-wise product purchase frequency becomes a competitive baseline for recommending grocery products, as users' most loyal products can be 'memorized' based on these statistics. Of course, such a baseline lacks generalization power since it is unable to capture product semantics. Thus an appropriate algorithm for product recommendation should balance item-to-item complementarity, user-to-item compatibility, and users' product loyalty.

Summary of Contributions

On the basis of product and user representations, we propose a novel algorithm **adaLoyal** for personalized grocery recommendation. Our method is capable of adaptively balancing users' "must-buy" products with preferences inferred from the low-dimensional representations.

Based on the quantitative results from experiments, we demonstrate that by applying **adaLoyal**, performance of a variety of embedding learning methods can be dramatically improved. The effectiveness of product loyalty estimated from **adaLoyal** can be validated in our qualitative analysis as well.

We also reveal modeling users' product loyalty and repeated purchases is critical in grocery product recommendation tasks, and such loyalty varies across different users, store types and product categories.

3.4.1 Related Work

Brand loyalty and repeat purchasing behavior in grocery shopping have been theoretically and empirically studied in the areas of economics and psychology [38, 73, 134]. Although conceptually different, loyalty and repeat consumption correlate to each other [73], and such loyalty varies across products and categories [38, 134]. These studies also motivate us to extend the concept of 'loyalty' to large-scale product recommender systems. Recently, a 'Wide&Deep' approach was proposed to address both *memorization* and *generalization* issues in Google Play application recommendations [32]. Rather than modeling user-to-product loyalty, however, they seek to memorize frequently co-occurring products or features. Another line of relevant work includes modeling repeat consumption in online activities including video/music streaming [10, 19, 29, 30, 83, 91] and e-commerce [97]. The lifetime of an item in this context is relatively short and users' interest highly depends on recency, which is different from empirical findings from grocery shopping where the decline in loyalty is relatively small over time [38]. Therefore, new techniques need to be developed to handle the dynamics of this specific domain.

3.4.2 Methodology

We find that in grocery baskets, a number of shoppers have their own 'must-buy' products. A preliminary analysis of 'must-buy' products is provided in Figure 3.8. Such repeat purchases could easily be measured based on user-wise item purchase frequency but may not be captured by low-dimensional product and user representations. Therefore, we introduce an algorithm **adaLoyal** to adaptively combine these two components and estimate users' product loyalty over time.

We start with a Bayesian view and then gradually build **adaLoyal** based on this principle. Specifically, for a user u, a product i and a transaction t, we have two predictive models for this basket: the prior purchase probability $p_{i,u}$, and the empirical item purchase frequency $q_{i,u}^{(t-1)}$ up

to the given transaction t.[19] We introduce a latent loyalty indicator $L_{i,u}$, which acts as a 'switch' such that $L_{i,u}$ (u is loyal to i) causes predictions to be generated from frequency only, while $\neg L_{i,u}$ (u is not loyal to i) causes the probability to be estimated from representations. Then the ultimate item purchase probability can be generated from the following probabilistic mixture:

$$P(C_{i,u}^{(t)}=1) = P(L_{i,u})\underbrace{P(C_{i,u}^{(t)}=1|L_{i,u})}_{\text{frequency model: } q_{i,u}^{(t-1)}} + P(\neg L_{i,u})\underbrace{P(C_{i,u}^{(t)}=1|\neg L_{i,u})}_{\text{representation model: } p_{i,u}},$$

where $C_{i,u}^{(t)}=1$ indicates that i is purchased by user u in transaction t. On the other hand, given $C_{i,u}^{(t)}$, the posterior distribution of this loyalty indicator is:

$$P(L_{i,u}|C_{i,u}^{(t)}) = \frac{P(L_{i,u})\ P(C_{i,u}^{(t)}|L_{i,u})}{P(L_{i,u})\ P(C_{i,u}^{(t)}|L_{i,u}) + P(\neg L_{i,u})\ P(C_{i,u}^{(t)}|\neg L_{i,u})}. \tag{3.22}$$

Inspired by this posterior distribution, we seek to estimate a weight (i.e., the product 'loyalty') $l_{i,u}^{(t)} \in [0,1]$ which results in an ultimate product purchase probability for the next basket:

$$\tilde{p}_{i,u}^{(t+1)} = l_{i,u}^{(t)} q_{i,u}^{(t)} + (1-l_{i,u}^{(t)}) p_{i,u}. \tag{3.23}$$

i.e., $l_{i,u}^{(t)}$ is used to approximate $P(L_{i,u})$ adaptively. Finally we propose the **adaLoyal** algorithm as follows. We scan a user's transaction logs chronologically: (1) if a new product is observed, we activate its corresponding loyalty $l_{i,u}^{(t)}$ and set it to be a given initial value l_0; (2) if a product has been purchased before, $l_{i,u}^{(t)}$ is updated based on the posterior distribution of the loyalty indicator.

The pseudo-code of **adaLoyal** is given in algorithm 1 and the specific update rules for $l_{i,u}^{(t)}$ are provided in Eq. (3.24) and Eq. (3.25). In general, if a user starts repeatedly consuming a particular product beyond our expectation (i.e., it diverges from the representation model), the

[19] $q_{i,u}^{(t-1)}$ is defined as the number of purchases of product i divided by the total number of products purchased up to the given transaction t.

Algorithm 1 Pseudo-code of **adaLoyal**

Input: $p_{i,u}$, $q_{i,u}^{(t)}$, $C_{i,u}^{(t)}$, l_0.
Output: $\tilde{p}_{i,u}^{(t)}$, $l_{i,u}^{(t)}$.
for each user u, each item i, each transaction t **do**
  if $q_{i,u}^{(t-1)} = 0$ **then**
    // current item has not been purchased before
    assign $\tilde{p}_{i,u}^{(t)} = p_{i,u}$
    assign $l_{i,u}^{(t)} = l_0$, if $C_{i,u}^{(t)} = 1$; $l_{i,u}^{(t)} = NA$, otherwise.
  else
    // loyalty of current item has been activated
    assign $\tilde{p}_{i,u}^{(t)} = l_{i,u}^{(t-1)} q_{i,u}^{(t-1)} + (1 - l_{i,u}^{(t-1)}) p_{i,u}$
    if $C_{i,u}^{(t)} = 1$ **then**
      assign $l_{i,u}^{(t)} = \frac{l_{i,u}^{(t-1)} q_{i,u}^{(t-1)}}{l_{i,u}^{(t-1)} q_{i,u}^{(t-1)} + (1 - l_{i,u}^{(t-1)}) p_{i,u}}$ (3.24)
    else
      assign $l_{i,u}^{(t)} = \frac{l_{i,u}^{(t-1)} (1 - q_{i,u}^{(t-1)})}{l_{i,u}^{(t-1)} (1 - q_{i,u}^{(t-1)}) + (1 - l_{i,u}^{(t-1)})(1 - p_{i,u})}$ (3.25)
    end if
  end if
end for

corresponding product loyalty will increase and its purchase frequency will be emphasized in the final prediction.

Note that the same algorithm can also be applied to within-basket recommendation, where the ultimate product purchase probability for user u given product j in the basket can be estimated as $\tilde{p}_{i,uj}^{(t+1)} = l_{i,u}^{(t)} q_{i,u}^{(t)} + (1 - l_{i,u}^{(t)}) p_{i,uj}$.

3.4.3 Experiments

We use the same datasets and the same settings (i.e. next-basket and within-basket recommendation tasks) as in Section 3.3.3 to evaluate the recommendation boost from **adaLoyal**.

As a preliminary analysis of users' product loyalty in grocery shopping, we explore the distribution of the item purchase frequency of each user's most favored product, $q_{\max}$.[20] In Figure 3.8 we split users into three groups: $q_{\max} \in (0, 0.1], (0.1, 0.5], (0.5, 1]$ to show the

[20]For users who have at least 10 transactions in each dataset, we calculate the user-wise item purchase frequency and find the maximum $q_{\max}$ for each user (i.e., the purchase frequency of the user's most favored product).

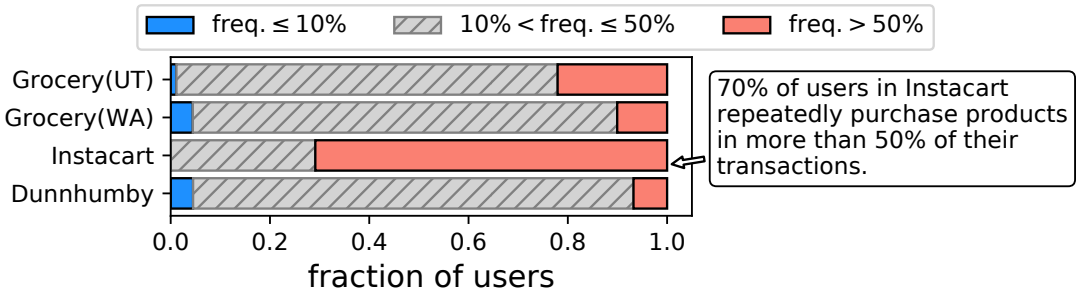


Figure 3.8: Distribution of the purchase frequency of each user's most favorite product.

distribution. We notice that the fraction of users where $q_{\max} \leq 0.1$ is limited, which means most users exhibit some repeat consumptions. Moreover, different from other datasets, around 70% of users in ***Instacart*** have products which are repeatedly purchased in more than half of their transactions. A possible explanation could be that ***Instacart*** is collected from regular shoppers on an online grocery shopping platform, where users may repeatedly seek the same products for efficiency rather than browsing and exploring as in physical stores.

Quantitative Results

We select the initial loyalty value l_0 based on the performance on the validation set.[21]. Detailed results on the next-basekt recommendation task are included in Table 3.12a. As an incremental algorithm, we also compare **adaLoyal** and the **BPR** loss. To make a fair comparison, we extract item and user embeddings from all representation learning methods and learn a weighted inner product between these two kinds of embeddings as a ranking score. The associated weights are learned by applying the same pairwise ranking loss (i.e., the **BPR** loss [139]). Note we adopt the same supervised learning protocol here but only need to learn K parameters,[22] where K is the dimensionality of latent embeddings.

We notice that although **BPR-MF** and **FPMC** outperform most representation learning methods without incorporating product loyalty, by applying the same supervised learning loss

[21] $l_0 \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$

[22] In practice, we may need to learn item bias terms as well.

function and learning minimal parameters, these representation learning methods can achieve comparable results on some datasets in terms of AUC and better top-biased ranking performance on all the datasets in terms of NDCG (see the difference between 'BPR-MF' and '+BPR' in Table 3.12a). Note that user-wise **itemPop** yields strong performance based on NDCG but poor performance based on the overall ranking metric AUC, as such a frequency-based method is not capable of capturing basket semantics and lacks generalization power for future purchases. Nevertheless, its top-biased performance is stronger than **BPR-MF** and **FPMC** which directly optimize a ranking metric. A possible reason could be that users' favoritism to some products is difficult to model using latent low-dimensional representations but easy to memorize based on purchase frequency. For the same reason, the performance of all embedding learning methods is significantly boosted in terms of both AUC and NDCG by applying **adaLoyal** to effectively combine these two models (see the difference between the first row and '+adaLoyal' in each group of Table 3.12a). Similar patterns can be observed for the within-basket recommendation task (Table 3.12b) as well.

We further explore the improvement from **adaLoyal** on users' repurchases (i.e., products that have be purchased in the given user's training transactions) and new purchases (i.e., products that have not been purchased by the user before the test transaction). Results on the ***Dunnhumby*** and ***Instacart*** datasets are provided in Figure 3.9. We find that by applying **adaLoyal**, recommendations on repurchases will be boosted to the upper bound provided by userwise **itemPop**. Doing so only sacrifices limited performance when generalizing to new purchases. This implies that the algorithm benefits from both the frequency model and universal embeddings by successfully distinguishing 'must-buy' and 'on-demand' products. The effectiveness of estimated product loyalties will be further validated in the subsequent case studies.

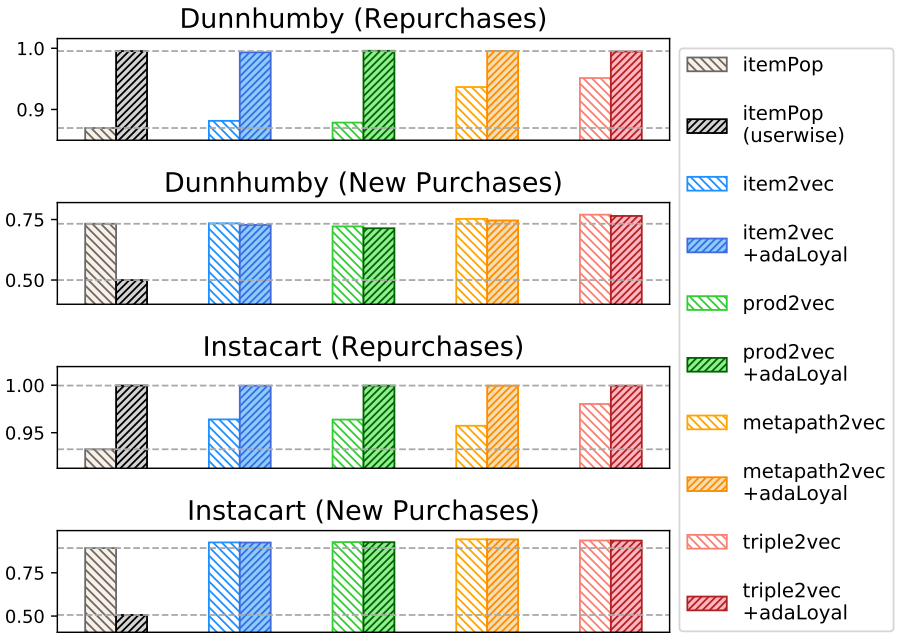


Figure 3.9: Results for repurchased products and newly purchased products in next-basket recommendation tasks on the ***Dunnhumby*** and ***Instacart*** datasets (in terms of AUC).

3.4.4 Case Studies

We further conduct qualitative analysis to explore the variety of product loyalty across different users and categories on the ***Instacart*** dataset.

Here we explore product loyalty estimated from **adaLoyal+triple2vec** in detail. We fix the initial loyalty $l_0 = 0.5$, and collect the estimated loyalties $l_{i,u}^{(t)}$ for each user at the final timestamp in the training set. Then we calculate the average value of product loyalties for each user and provide its distribution across different datasets in Figure 3.10. In this figure, we find that *Instacart* is more "loyalty"-dominated compared with physical grocery stores.

We also provide concrete transaction examples in Table 3.10 to illustrate how users behave differently from each other in terms of product loyalty. We find a few users who exhibit strong product loyalty similar to *User A* in Table 3.10, i.e., they order certain products in every transaction. However, most users' shopping patterns are better reflected by *User B*'s transactions, where they have strong preferences on some products but occasionally switch brands (e.g. Taboule

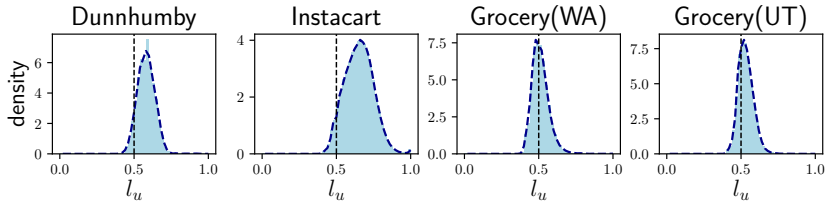


Figure 3.10: Histograms of user's product loyalty across different datasets, where l_u represents the average product loyalty of each user with the same initialization $l_0 = 0.5$.

Table 3.10: Baskets from users with the same number of transactions, but different average product loyalties in the *Instacart* dataset.

User A ($l_u = 1.00$)
Sparkling Water, Bottles
Sparkling Water, Bottles
Sparkling Water, Bottles
Sparkling Water, Bottles
Sparkling Water, Bottles
Sparkling Water, Bottles

User B ($l_u = 0.57$)
Spinach Artichoke Dip, Taboule Salad, ...
Packaged Grape Tomatoes
Bag of Organic Bananas, Taboule Salad
Fuji Apples, Seedless Cucumbers, ...
Bag of Organic Bananas, Sweet Kale Salad Mix
Spinach Artichoke Dip, Seedless Red Grapes, ...

User C ($l_u = 0.37$)
Olive Oil Soap, Citrus Castile Soap, Peppermint Castile Soap ...
Coconut Chips – Sea Salt, Coconut Chips – Original ...
Compostable Forks
Grunge Buster Grout And Tile Brush
Pumpkin Seed Cheddar Crispbreads, Seedlander Crispbreads
Zinc Target Mins 50 Mg Gluten Free Tablets

Salad and Sweet Kale Salad Mix). Different from these two kinds of users, product-*unloyal* consumers prefer to explore the store rather than sticking to particular products. For example, *User C* in Table 3.10 has hardly any repeated product consumptions, rather they buy different products with the same function (e.g. various soaps, coconut chips, crispbreads) in the same basket.

Next we calculate the average product loyalties for each department and category. Based on these statistics, the five most loyal and unloyal departments/categories are listed in Table 3.11a. Note that the distribution of this absolute loyalty estimation potentially correlates to the de-

Table 3.11: The five most loyal/unloyal departments and categories in the *Instacart* dataset.

(a) Department/category ranking based on the raw loyalty score l_u.

loyal dept.	loyalty	unloyal dept.	loyalty	loyal cat.	loyalty	unloyal cat.	loyalty
pets	0.64	pantry	0.52	milk	0.68	kitchen supp.	0.45
dairy/eggs	0.63	personal care	0.52	eggs	0.68	baking decor.	0.45
beverages	0.61	other	0.52	water/seltzer	0.66	spices	0.46
bakery	0.61	household	0.53	energy drinks	0.65	first aid	0.46
breakfast	0.61	international	0.54	lactose free	0.65	beauty	0.48

(b) Department/category ranking based on the relative score $z_l - z_f$ (i.e., difference between z-scores of loyalty and frequency).

loyal dept.	z_l	z_f	$z_l - z_f$	unloyal dept.	z_l	z_f	$z_l - z_f$
pets	1.638	-0.826	2.464	meat/seafood	0.326	2.353	-2.027
dairy/eggs	1.314	-0.734	2.048	international	-1.117	0.857	-1.973
bulk	-0.395	-1.606	1.210	pantry	-1.768	-0.223	-1.544
babies	0.509	-0.606	1.115	beverages	0.951	1.499	-0.547
alcohol	0.100	-0.717	0.817	dry goods/pasta	-0.349	0.165	-0.514
loyal cat.	z_l	z_f	$z_l - z_f$	unloyal cat.	z_l	z_f	$z_l - z_f$
mint gum	0.933	-0.356	1.289	fresh fruit	0.408	4.817	-4.408
dog food care	1.215	0.067	1.148	fresh vegetables	0.255	4.234	-3.980
granola	0.986	-0.158	1.144	pack. veg./fruits	0.525	3.006	-2.481
energy drinks	1.709	0.656	1.053	spices	-2.377	-0.618	-1.759
eggs	2.264	1.213	1.051	fresh herbs	-0.932	0.464	-1.396

mand/necessity of a category. For example, we observe users are loyal to products such as "milk" and "eggs" (i.e., they repeatedly purchase a specific product in these categories) while these categories are highly in-demand and need to be purchased frequently in our daily life. To further investigate the loyalty/necessity relationship we normalize both product loyalties and department/category purchase frequencies into z-scores.[23] Then we calculate the difference between their z-scores to investigate how surprisingly loyal and unloyal is a particular department/category. We provide the five most relatively loyal and unloyal departments/categories in Table 3.11b, where we observe that users are relatively unloyal to fresh fruits, vegetables, meat, and spices but loyal to mint gums, granola, baby products, alcohol, and pet products.

[23]The z-score is defined as: $(x - mean(x))/sd(x)$

3.5 Conclusions and Future Work

In Section 3.2, we systematically studied the problem of modeling consumer preferences and price sensitivities, and proposed a nested feature-based matrix factorization framework to support personalized and scalable recommendation and demand systems. We verified that the proposed model is capable of providing high quality preference predictions and specific price elasticity can be appropriately estimated for each shopping trip. By applying the proposed framework on two real-world datasets, we provided economic insights which may benefit both data mining and economics communities.

Particularly, we noticed that price affects product choice but has limited effects on category purchase or product quantity, which means coupons are primarily effective "within category". Price sensitivity in large-scale systems is an important problem and a number of possible topics can be explored along this trajectory. For example, temporally-aware models could be developed to allow long-term purchase patterns to be carefully studied. Cross elasticity has been introduced but not completely explored in this work; this could be studied in detail in future work where not only product substitution but product complementarity could be modeled. In the context of hybrid recommender and demand systems, we have so far only studied consumer behavior in this work, but the optimization strategies could be adapted to generate personalized coupons.

In Section 3.3 and Section 3.4, we investigated grocery shopping behavior and observed three important patterns in users' baskets—complementarity between products, compatibility between users and products, and users' product loyalty. We proposed a new representation learning method, **triple2vec**, to holistically leverage complementarity and compatibility, and designed a novel algorithm **adaLoyal** for product recommendation by adaptively balancing universal product embeddings and users' product loyalty over time. We demonstrated their effectiveness through quantitative and qualitative results on two public and two proprietary grocery datasets.

The idea of complementarity, compatibility, and loyalty is not limited to grocery shopping but can be widely applied on other domains, especially those with repeated consumptions (e.g. music streaming). It would also be interesting to extend **adaLoyal** to be a Bayesian reinforcement learning framework, which could learn the product loyalty and update the recommendations in an interactive environment.

3.6 Acknowledgements

This chapter is based on the materials as they appear in the *International Conference on World Wide Web*, 2017 ("Modeling Consumer Preferences and Price Sensitivities from Large-Scale Grocery Shopping Transaction Logs," Mengting Wan, Di Wang, Matt Goldman, Matt Taddy, Justin Rao, Jie Liu, Dimitrios Lymberopoulos, and Julian McAuley), and the *ACM Conference on Information and Knowledge Management*, 2018 ("Representing and Recommending Shopping Baskets with Complementarity, Compatibility, and Loyalty," Mengting Wan, Di Wang, Jie Liu, Paul Bennett, and Julian McAuley). The dissertation author was the primary investigator and author of these papers.

Table 3.12: Detailed results on recommendation tasks ($K = 32$).

(a) AUC and NDCG on next-basket recommendation.

	Dunnhumby		***Instacart***		***Grocery(WA)***		***Grocery(UT)***	
Method	AUC	NDCG	AUC	NDCG	AUC	NDCG	AUC	NDCG
itemPop	0.799	0.129	0.918	0.145	0.809	0.130	0.854	0.137
+userwise	0.732	0.175	0.773	0.273	0.591	0.150	0.607	0.141
BPR-MF	0.861	0.136	0.964	0.161	*0.831*	0.136	0.862	0.139
FPMC	0.853	0.137	0.963	0.163	0.821	0.139	0.865	0.139
item2vec	0.801	0.132	0.945	0.111	0.794	0.138	0.822	0.108
+BPR	0.851	0.145	0.964	0.188	0.804	0.141	0.846	0.138
+adaLoyal	0.853	0.181	0.963	0.270	0.805	0.174	0.858	0.133
prod2vec	0.796	0.119	0.945	0.115	0.790	0.137	0.826	0.119
+BPR	0.850	0.144	0.964	0.183	0.807	0.144	0.852	0.140
+adaLoyal	0.848	0.154	0.964	0.273	0.803	0.175	0.853	0.138
m.2vec	0.838	0.144	0.954	0.125	0.810	0.126	0.846	0.113
+BPR	0.854	0.153	0.959	0.189	0.809	0.145	0.856	0.147
+adaLoyal	0.862	*0.182*	0.967	0.269	0.820	0.174	0.874	0.149
triple2vec	0.852	0.129	0.959	0.128	0.817	0.137	0.848	0.124
+BPR	0.861	0.142	0.962	0.186	0.819	0.149	0.854	0.144
+adaLoyal	*0.870*	0.166	*0.968*	*0.277*	0.830	*0.176*	*0.875*	*0.152*

(b) AUC and NDCG on within-basket recommendation.

	Dunnhumby		***Instacart***		***Grocery(WA)***		***Grocery(UT)***	
Method	AUC	NDCG	AUC	NDCG	AUC	NDCG	AUC	NDCG
itemPop	0.795	0.129	0.918	0.145	0.809	0.131	0.854	0.137
+userwise	0.730	0.174	0.773	0.272	0.590	0.149	0.606	0.141
item2vec	0.831	0.145	0.941	0.116	0.835	0.159	0.868	0.117
+adaLoyal	0.878	0.183	0.965	0.273	*0.849*	0.190	0.883	0.140
prod2vec	0.803	0.121	0.941	0.125	0.820	0.148	0.850	0.125
+adaLoyal	0.856	0.173	0.965	0.273	0.836	0.184	0.866	0.142
m.2vec	0.834	0.144	0.944	0.125	0.810	0.126	0.846	0.113
+adaLoyal	0.858	0.182	0.960	0.269	0.820	0.173	0.874	0.146
triple2vec	0.864	0.145	0.960	0.127	0.830	0.153	0.869	0.132
+adaLoyal	*0.879*	*0.185*	*0.970*	*0.279*	0.843	*0.191*	*0.885*	*0.157*

Chapter 4

Modeling Structures of Heterogeneous Consumer Activities

4.1 Introduction

User feedback in recommender systems is usually classified into two categories: 'explicit' feedback—where users directly express their preferences (e.g. ratings), and 'implicit' feedback—where users indirectly reveal their interests through actions (e.g. clicks). These two paradigms have long been studied as two separate topics and different techniques have been developed to address each of their distinct properties.

Beyond the narrow definitions of explicit versus implicit feedback, we notice that multiple types of user feedback are abundant in many real-world information systems. For example, users' views, clicks, purchases and rating scores are commonly available on e-commerce platforms. All of these signals reflect (or imply) users' interests regarding items from different perspectives. Although there has been a line of work where the connections between implicit and explicit interactions are considered [75, 76, 88, 103, 124, 126, 127], most focus on improving numeric rating predictions by leveraging other signals as auxiliary information. These studies motivate us

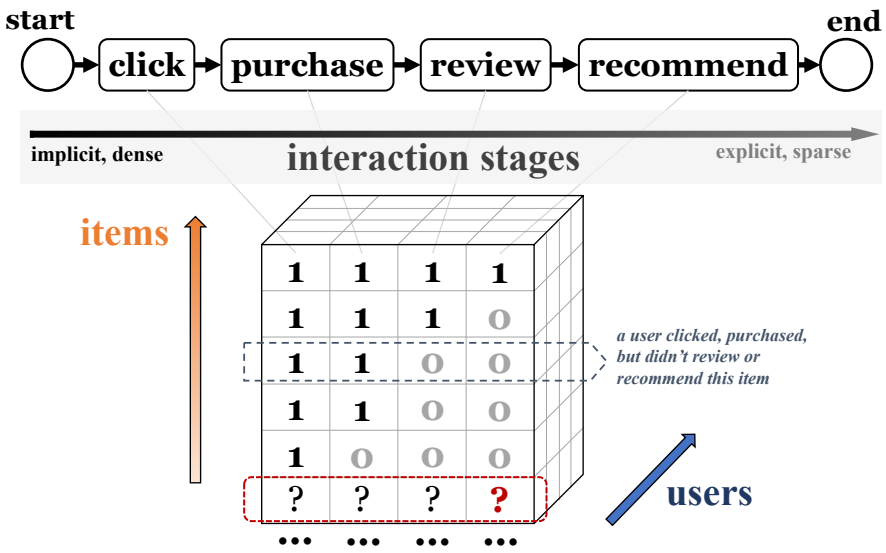


Figure 4.1: Illustration of monotonic behavior chains and the associated item recommendation problems.

to bridge the gap between implicit and explicit signals, but our primary goal is to build a unified recommendation framework for a more general purpose, where several types of user feedback can be simultaneously considered regardless of their specific semantics. Specifically we are interested in (1) how to properly represent a spectrum of users' responses; and (2) how to efficiently harness the connections among these interactions and provide personalized item recommendations.

User-Item Interactions as Behavior Chains

We typically observe relatively a few explicit responses (such as numeric rating scores), but abundant implicit feedback such as 'click' and 'purchase' actions. Although these interactions are heterogeneous in terms of both representations and data distributions, they can be aligned on what we refer to as 'behavior chains.' As shown in Figure 4.1, different user-item interactions in e-commerce systems can be encoded as binary states on a chain, which semantically represents if a user clicks, purchases, reviews or recommends, (e.g. a rating score larger than some threshold) the item. Specifically, the highlighted vector $(1,1,0,0)$ in Figure 4.1 encodes that a user clicked,

purchased, but didn't review or recommend a product. By following the links on these chains, users gradually 'activate' interaction stages which increasingly imply more explicit preferences toward items. Such representations provide us with not only a template to unify different types of interactions but also a prototype to simulate users' decision-making processes.

Monotonicity on Behavior Chains

One notable property of such behavior chains is their *monotonicity*. That is, once a user decides to 'stop' at a stage, then the subsequent interactions by definition will not be observed. For instance, we cannot observe a user's 'review' or 'recommend' actions if the item was not purchased by this user. By properly leveraging these monotonicity constraints, we hope to distinguish critical versus nonessential information. For instance, 'not review' and 'not recommend' are nonessential (i.e., already implied) given that the item was not purchased by the user. Because of this structure, for each user we can define a binary matrix on these behavior chains, which starts with *the most implicit* (and *densest*) responses, and ends with *the sparsest* but *the most explicit* responses. In each row of this matrix, elements are monotonically non-increasing from left to right.

Recommendation on Monotonic Behavior Chains

Based on the above representations, we can describe our primary goal in this paper as follows:

- **Goal:** *Given historical observations of users' behavior chains, we seek to estimate their responses toward unobserved items by appropriately leveraging the monotonicity assumption implied by the data.*

Note for each interaction stage on a behavior chain, we are able to define an associated one-class recommendation problem. We regard the recommendation performance on the most explicit (i.e., the last) stage as our primary evaluation criterion but also investigate the performance with

respect to the full interaction spectrum.

Summary of Contributions

We observe a common scenario where multiple types of user-item interactions can be aligned on a monotonic behavior chain, and propose a unified item recommendation problem based on this representation.

We propose a new algorithm—**chainRec** which effectively models multiple types of interactions and efficiently exploits the monotonicity among these actions. In particular, we design a scoring function on top of users' behavioral intentions in order to make use of all types of responses, explicitly model users' target intents, and preserve the monotonic constraints in the resulting user preference scores. We also develop a new optimization criterion which takes advantage of the monotonicity and automatically focuses on the most critical information in users' feedback data.

We evaluate the model on five different real-world datasets where our experiments indicate the proposed algorithm substantially outperforms baselines.

We contribute a new large-scale dataset (of book reviews) for this problem. It contains information from more than 200 million user-item interactions and covers four different interaction types (shelve – read – rate – recommend).

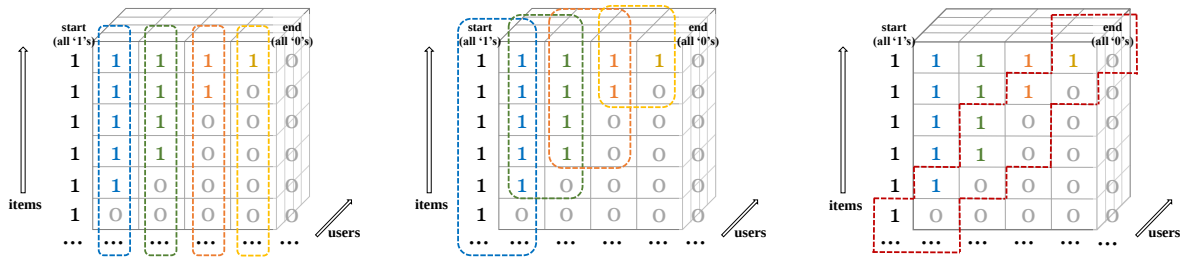
4.2 Related Work

Traditional item recommendation systems often rely on a suite of collaborative filtering techniques to learn from explicit feedback such as rating scores. Typical model-based techniques include matrix factorization (**MF**) methods [90], which seek to learn item and user embeddings and use the inner product to approximate observed ratings. As providing explicit feedback often requires additional cognitive effort [51], these interactions may be *sparse* or unavailable

in real-world scenarios. In such cases, the above **MF** methods can be extended to model the more abundant implicit signals that are disclosed via users' observable actions such as clicks and purchases [72, 123, 139]. In order to address the one-class property of this setting (i.e., only positive instances can be observed), several approaches including the pairwise ranking method **BPR** [139] and the pointwise optimization method **WRMF** [72, 123] have been proposed.

Although explicit and implicit data are commonly studied as two separate topics, there are several studies that seek to connect these signals [75, 103, 124, 126, 127]. These methods are summarized in a recent survey [76]. Specifically in the music recommendation domain, a positive correlation between implicit (e.g. play counts) and explicit feedback (e.g. ratings) has been found; regression models thus can be built to predict users' rating scores from implicit signals [126, 127]. In addition, a factorized neighborhood model was proposed to directly estimate users' ratings where implicit signals are used to locate and regularize the neighborhood items [88]. Furthermore, several methods have been proposed to jointly factorize users' rating scores and implicit responses with shared user and item latent factors [103, 124]. Unlike these methods which specifically handle numeric (i.e., 'star') ratings and regard other feedback as side-information, we seek to build a framework where several types of (binary) user feedback can be aligned without giving special treatment to rating scores.

In a recent study [168], tensor decomposition techniques are applied to model users' different types of activities and negative sampling strategies are introduced to address the one-class problem. Another line of relevant work includes session-based recommendation and modeling 'micro-behavior' in each user session [61, 69, 95, 162, 181]. These methods differ from each other in precise details, but they typically focus on embedding micro-behaviors within each user session (e.g. views, clicks, dwell time) into a predictive framework (e.g. a recurrent neural network) to estimate the users' next actions. Although all of the above methods seek to fuse users' activities, neither of them explores the 'strength' of each type of signal or the potential monotonic dependencies among these activities.



(a) sliceOpt (b) condOpt (c) edgeOpt

Figure 4.2: Illustration of different optimization criteria.

4.3 Problem Definition and Preliminary Learning Strategies

In this section, we formally define monotonic behavior chains and investigate several preliminary learning strategies for the proposed item recommendation problem.

Suppose for a user u and an item i, we have labels for a chain of user-item interactions $\boldsymbol{y}_{ui} = [y_{ui,1}, \ldots, y_{ui,L}]^T$, where L is the number of interaction stages and $\forall l = 1, \ldots, L$:

$$y_{ui,l} = \begin{cases} 1, & \text{if the interaction } (u,i) \text{ at stage } l \text{ is observed;} \\ 0, & \text{otherwise.} \end{cases}$$

That is if a user performs an action (e.g. click) on an item, it is regarded as a 'positive' instance; otherwise, as in traditional one-class settings, rather than simply treating the non-click as 'negative,' it could be that the user is simply not aware of the item, or the cost is too high (etc.). Typically we expect items interacted with by a user to be ranked higher than other items in the final recommendation. These behavior chains $\{\boldsymbol{y}_{ui}\}$ are said to be *monotonic* if $y_{ui,1} \geq y_{ui,2} \geq \ldots \geq y_{ui,L}$, $\forall u, i$. To simplify notation at boundaries, we assume two 'pseudo' stages $l = 0$ and $l = L+1$, where we always have $y_{ui,0} = 1$ and $y_{ui,L+1} = 0$, $\forall u, i$.

Then recommendation problems on these monotonic behavior chains can be formulated as estimating the ranking scores of unobserved items (i.e., items that a user has never interacted with) at each stage, where the same underlying ranking mechanisms can be used to approximate

the observed feedback $y_{ui,l}$.

Learning Preferences Independent of Stages

A naïve approach to solve our problem would be to ignore inter-stage dependencies and simply learn preference/ranking models for each stage independently. A representative objective function for stage l would be the *pointwise* binary cross-entropy (e.g. **LogisticMF** [80], **NeuMF** [67]):

$$-\sum_{u,i}\Big(y_{ui,l}\log\sigma(s_{ui,l})+c_{ui,l}(1-y_{ui,l})\log(1-\sigma(s_{ui,l}))\Big) \tag{4.1}$$

where $s_{ui,l}$ is u's preference ranking score regarding item i, $\sigma(\cdot)$ is the sigmoid function and $c_{ui,l}$ is a customized weight to balance positive and negative observations. In practice, c_{ui} can be implemented through sampling techniques during training [67, 123]. That is for each positive (u,i) pair in $I^+_{u,l}=\{i|y_{ui,l}=1\}$, we can sample N items with which the user did not interact at stage l, compose a 'balanced' negative itemset $I^-_{u,l}$, and update the binary cross-entropy loss function:

$$-\sum_{u}\Big(\sum_{i\in I^+_{u,l}}\log\sigma(s_{ui,l})+\sum_{i'\in I^-_{u,l}}\log(1-\sigma(s_{ui',l}))\Big). \tag{4.2}$$

Another popular objective function is the *pairwise* ranking loss (e.g. **BPR** [139]):

$$-\sum_{u,i\in I^+_{u,l},i'\in I^-_{u,l}}\log\sigma(s_{ui,l}-s_{ui',l}), \tag{4.3}$$

which seeks to maximize a pairwise difference between observed positive and unobserved 'negative' instances.

Here independent parameters are applied to model the ranking scores $s_{ui,l}$ for different stages; one popular underlying approach is the latent factor model:

$$s_{ui,l}=b_{0,l}+b_{i,l}+b_{u,l}+\langle\boldsymbol{\gamma}_{i,l},\,\boldsymbol{\gamma}_{u,l}\rangle, \tag{4.4}$$

where for each stage l, $b_{0,l}$ is the global offset, $b_{i,l}, b_{u,l}$ are item and user biases, and $\boldsymbol{\gamma}_{i,l}, \boldsymbol{\gamma}_{u,l}$ are K-dimensional embeddings to capture items' latent features and users' latent preferences toward these features. Here $\langle \cdot, \cdot \rangle$ denotes the inner product such that $\langle \boldsymbol{\gamma}_{i,l}, \boldsymbol{\gamma}_{u,l} \rangle$ captures the 'compatibility' between user u and item i on stage l.

Learning Preferences Jointly on Different Stages

Note that the above models ignore the underlying relationships among different interaction stages. Given that all of these interactions ought to reflect users' preferences from different perspectives, we can extend the assumption applied in existing studies [103, 124] that better item and user representations could be learned by jointly modeling different types of interactions. This results in a joint objective function that extends Eq. (4.2):

$$-\sum_{u,l} \Big(\sum_{i \in I^+_{u,l}} \log \sigma(s_{ui,l}) + \sum_{i' \in I^-_{u,l}} \log(1 - \sigma(s_{ui',l})) \Big). \tag{4.5}$$

Critically, the preference score $s_{ui,l}$ can be modeled by *shared* item and user embeddings, combined with a set of stage-specific weights $\boldsymbol{\gamma}_l$ on different latent dimensions, i.e.,

$$s_{ui,l} = b_0 + b_i + b_u + \langle \boldsymbol{\gamma}_l, \ \boldsymbol{\gamma}_i \circ \boldsymbol{\gamma}_u \rangle \,, \tag{4.6}$$

which is equivalent to a variant of the **CP/PARAFAC** tensor decomposition framework [24]. Here $\circ$ denotes the Hadamard product.

Note that together with the above learning strategies, optimization criteria of these approaches focus on the estimations within each vertical 'slice' (i.e., each stage l) of the observation matrices, but do not involve any horizontal connections among slices such as the monotonicity we discussed above. We thus refer them as 'slicewise' optimizations (**sliceOpt**, see Figure 4.2a).

Learning Preferences Conditioned on Previous Stages

We next aim to explore the 'monotonicity' property of these behavior chains. An obvious assumption would be that any observations of an interaction stage should be conditioned on the presence of the previous stage (e.g. a 'purchase' action should be conditioned on the presence of a 'click' action). Therefore instead of approximating the marginal probability of each stage directly, we seek to model the conditional probability of the behavior 'escalation' from a weaker to a stronger interaction. Specifically we consider the following conditional optimization criterion (**condOpt**, see Figure 4.2b):

$$-\sum_{u,l}\sum_{i\in I^{+}_{u,l-1}}\Big(y_{ui,l}\log p_{ui,l|l-1}+c_{ui,l}(1-y_{ui,l})\log(1-p_{ui,l|l-1})\Big) \tag{4.7}$$

where $p_{ui,l|l-1}$ is shorthand for $P(y_{ui,l}=1|y_{ui,l-1}=1)$; similar sampling techniques as in Eq. (4.2) can be used. Suppose we have $p_{ui,l|l-1}=\sigma(\delta_{ui,l})$. Then $\delta_{ui,l}$ can be factorized using stage-independent item/user embeddings (Eq. (4.4)) or shared item/user embeddings (Eq. (4.6)). We use the joint probability

$$s_{ui,l} \coloneqq P(y_{ui,1}=\ldots=y_{ui,l}=1)=\prod_{l'=1}^{l} p_{ui,l'|l'-1}$$

as the preference ranking score for item recommendations at stage l. This preference score naturally inherits the monotonicity in the interaction labels, i.e., $s_{ui,1}\geq\ldots\geq s_{ui,L}$.

As shown in Figure 4.2b, this learning strategy gradually narrows its training scope by conditioning on previous observed interactions. As positive instances for explicit feedback are relatively scarce, following this conditional optimization criterion may lead to difficulty capturing behavior escalations that happen at later stages of behavior chains. On the other hand, this strategy can be regarded as being analogous to executing a sequence of 'AND' operators on the interaction chains. That is, in order to reach the most explicit stage, all of the behavior escalations have to be

'activated.' One potential drawback of this philosophy is that it tends to propagate failures to the subsequent stages. For example, it is difficult to interpret a user's relative preferences toward (say) an item that has been reviewed but not recommended versus an item that has been purchased but not reviewed. Combined with its limitation in handling data scarcity, the above learning strategy may have difficulty bypassing the noisy, implicit (and 'weakly' negative) data and thus fail to learn users' preferences accurately. Such an observation motivates us to develop new techniques to carefully model the internal logic behind the monotonicity of observed interactions.

4.4 The Proposed Algorithm

Based on the above investigation, we develop a new algorithm—**chainRec**—that exhibits the following properties: (1) it takes advantage of all stages of interactions to learn item/user representations for preference ranking; (2) the ultimate preference scores generated from the model explicitly preserve the monotonicity of the interaction matrix; (3) replacing the above 'AND' philosophy, we try to understand which interactions directly come from users' *intrinsic* behavior intentions, versus which are *derived* from (stronger) subsequent intentions. We use two techniques to achieve these properties: monotonic preference scoring functions and an edgewise optimization criterion.

Monotonic Scoring Function

We still model the marginal probability $P(y_{ui,l})$ but from a different prospective:

$$p_{ui,l} \coloneqq P(y_{ui,l} = 1) = \sigma(s_{ui,l}) = \frac{1}{1+\exp(-s_{ui,l})}, \tag{4.8}$$

where $s_{ui,l}$ is the preference score. Instead of directly decomposing the response $y_{ui,l}$, we introduce an additional layer for users' behavioral intentions. Here we use a similar **CP/PARAFAC** tensor

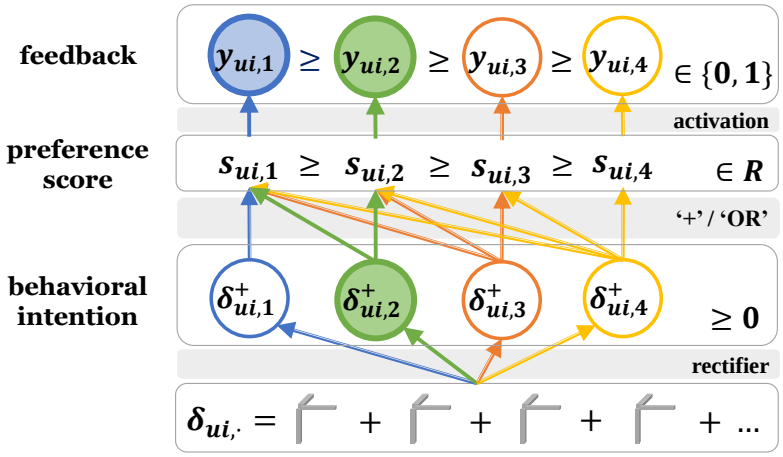


Figure 4.3: Illustration of our monotonic preference scoring function. In this example, only the behavioral intention $\delta^+_{ui,2}$ is activated. The observation $y_{ui,2} = 1$ directly comes from its activated associated intention $\delta^+_{ui,2}$, while $y_{ui,1} = 1$ is derived by its subsequent behavioral intention $\delta^+_{ui,2}$.

decomposition format as in Eq. (4.6) to factorize an intention score for each stage l:

$$\delta_{ui,l} = \langle \boldsymbol{\gamma}_l,\ \boldsymbol{\gamma}_i \circ \boldsymbol{\gamma}_u \rangle . \tag{4.9}$$

We then pass this intention score to a parametric rectifier such that a user's specific behavioral intention is activated if and only if this score exceeds zero:

$$\delta^+_{ui,l} = \frac{1}{\beta} \log(1 + \exp(\beta\, \delta_{ui,l})). \tag{4.10}$$

This rectifier becomes a softplus function when $\beta = 1$ and approximates a rectified linear unit (**ReLU**) as $\beta \to \infty$ (Figure 4.4). We assume $\beta \geq 1$ is a parameter which will be automatically learned during training.

On top of this layer, we assume that a user-item interaction stage can be observed if its associated *or* subsequent behavioral intentions are activated (as shown in Figure 4.3). We encode

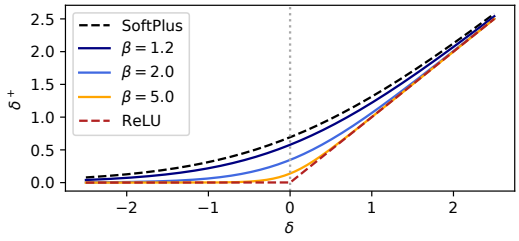


Figure 4.4: Example activation functions.

this soft logic in the final observations as follows:

$$s_{ui,l} = b_0 + b_i + b_u + \sum_{l'=l}^{L} \delta^+_{ui,l'}. \tag{4.11}$$

Note that the most explicit interaction can only be observed when its associated behavioral intention is activated. By modeling users' intentions in this format, both preference scores and the resulting probabilities preserve the monotonicity in observations $y_{ui,l}$, i.e., $s_{ui,l} - s_{ui,l+1} = \delta^+_{ui,l} \geq 0$.

Edgewise Optimization Criterion

In addition to this scoring function, we can encode the monotonicity of $y_{ui,l}$ into a probabilistic framework by enforcing these equivalent constraints

$$\begin{aligned}
&P(y_{ui,l'} = 1 | y_{ui,l} = 1) = 1, && \forall l' < l; \\
&P(y_{ui,l'} = 0 | y_{ui,l} = 0) = 1, && \forall l' > l; \\
&P(y_{ui,1} = 1, \ldots, y_{ui,l} = 1) = P(y_{ui,l} = 1), && \forall l; \\
&P(y_{ui,l+1} = 0, \ldots, y_{ui,L} = 0) = P(y_{ui,l+1} = 0), && \forall l.
\end{aligned} \tag{4.12}$$

These constraints help us prune the redundant information within the joint probability such that we can obtain the following reduced objective function on the 'edges' (i.e., two consecutive stages

Algorithm 2 chainRec

for each user u, and each item $i \in I_u^+$ **do**
  Locate the last positively interacted stage l^*_{ui}
  Update the associated parameters Θ_{ui} based on the gradients

$$\frac{\partial}{\partial \Theta_{ui}} \log p_{ui,l^*_{ui}}$$

  Sample N contrastive items based on the given sampling scheme
  for each contrastive item i' **do**
    Locate the last positively interacted stage $l^*_{ui'}$
    Update the associated parameters $\Theta_{ui'}$ based on the gradients

$$\frac{\partial}{\partial \Theta_{ui'}} \Big(\log \Big(1 - p_{ui',l^*_{ui'}+1} \Big) + \log p_{ui',\cap} \Big)$$

  end for
end for

where users exhibit different responses) of users' behavior chains (**edgeOpt**):

$$\sum_{u,i} \log P(y_{ui,1}, \ldots, y_{ui,L}) = \sum_{u,i} \log P(y_{ui,l^*_{ui}} = 1, y_{ui,l^*_{ui}+1} = 0) \tag{4.13}$$

where $l^*_{ui} = \arg\max_l \{y_{ui,l} = 1\}$ is the last stage at which the interaction (u,i) can be observed.[1]

As shown in Figure 4.2, **edgeOpt** differs from **sliceOpt** and **condOpt** in that it focuses on the most critical signals—observations at 'edges' of the behavior chains. The reason **edgeOpt** allows us to do this is the previous and the subsequent interactions are already implied by the monotonicity and guaranteed by applying monotonic scoring functions.

Notice that we are still facing the one-class problem, which means we generally trust the positive interactions $y_{ui,l^*} = 1$ but are not confident in unobserved 'negative' instances $y_{ui,l^*+1} = 0$. Therefore, similar to the weighting techniques used in previous one-class collaborative filtering studies [72, 80, 123], we seek to separate information contained in these two consecutive stages

[1] For brevity we may omit the subscript ui and use l^* in subsequent paragraphs.

from their joint probability and rebalance positive and negative instances. Specifically we have

$$P(y_{ui,l^*}=1, y_{ui,l^*+1}=0) = p_{ui,l^*}\,(1-p_{ui,l^*+1})\,p_{ui,\cap}, \tag{4.14}$$

and $p_{ui,\cap}$ denotes the (exponential of the) pointwise mutual information (**PMI**) between two consecutive stages on the edge

$$p_{ui,\cap} \coloneqq \frac{P(y_{ui,l^*}=1, y_{ui,l^*+1}=0)}{P(y_{ui,l^*}=1)\,P(y_{ui,l^*+1}=0)}.$$

By applying Eq. (4.12), Eq. (4.10) and Eq. (4.11), we have an explicit formula to calculate this information:

$$\begin{aligned} p_{ui,\cap} =& \frac{p_{ui,l^*} - P(y_{ui,l^*}=1, y_{ui,l^*+1}=1)}{p_{ui,l^*}\,(1-p_{ui,l^*+1})} = \frac{p_{ui,l^*} - p_{ui,l^*+1}}{p_{ui,l^*}\,(1-p_{ui,l^*+1})} \\ =& 1-\exp(-\delta^+_{ui,l^*}). \end{aligned} \tag{4.15}$$

Then we obtain the following rebalanced objective:

$$\begin{aligned} &\sum_u \Big(\sum_{i\in I_u^+} \log p_{ui,l^*} + \sum_{i'\in I} c_{ui',l^*} \big(\log\Big(1-p_{ui',l^*+1}\Big) + \log p_{ui',\cap} \big) \Big) \\ \approx &\sum_u \Big(\sum_{i\in I_u^+} \log p_{ui,l^*} + \sum_{i'\in \bar{I}_u} \big(\log\Big(1-p_{ui',l^*+1}\Big) + \log p_{ui',\cap} \big) \Big). \end{aligned} \tag{4.16}$$

We propose the following two sampling schemes to compose the above contrastive item set $\bar{I}_u$:

- **Uniform Sampling.** $c_{ui,l^*} \propto 1$. For each positive user-item pair (u,i) (i.e., $l_{ui}* > 0$), we uniformly sample N items regardless of their labels;
- **Stagewise Sampling.** $c_{ui,l^*} \propto \frac{|I^+_{u,l^*+1}|}{|I^+_{u,l^*}|}$. For each positive user-item pair (u,i), we sample N items based on their associated edge stage l^*_{ui}.

We briefly describe the proposed method **chainRec** in algorithm 2. Note that, in spite of the relatively complex derivation, the final algorithm is straightforward. We apply a standard ℓ_2

Table 4.1: Basic dataset statistics.

Dataset	#item	#user	#interaction	distribution of interactions	#inter. /#item	#inter. /#user
Steam	8,696	24,110	2,447,847	purchase (100.0%) play (64.0%) review (2.2%) recommend (2.0%)	281.49	101.53
YooChoose	19,034	509,126	2,292,077	click (100.0%) purchase (45.7%)	120.42	4.50
Yelp	119,340	1,005,382	4,731,170	review (100.0%) recommend (71.1%)	39.64	4.71
GoogleLocal	539,767	3,063,444	5,968,216	review (100.0%) recommend (85.0%)	11.06	1.95
Goodreads	1,561,465	808,749	225,394,930	shelve (100.0%) read (49.1%) rate (45.9%) recommend (32.0%)	144.35	278.70

regularizer[2] on item and user embeddings $\boldsymbol{\gamma}_i, \boldsymbol{\gamma}_u$ and **ADAM** [84] for optimization. As our primary goal is to rank unobserved items based users' most explicit preferences, we track the cross-entropy loss for the last stage L on a held-out validation set and stop training once it no longer decreases. All results are reported on the test set and all hyperparameters are selected based on the performance on the validation set.

4.5 Experiments

We evaluate **chainRec** and alternatives on five real-world datasets, where multiple types of user-item interactions are available. In particular we are interested in determining (1) whether recommendation performance for the sparsest (and thus 'most explicit') feedback can be improved by leveraging other types of interactions; (2) to what extent accounting for monotonicity can help with ranking performance; (3) whether recommendation performance on the dense (and implicit)

[2]The hyperparameter λ is selected from $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1.0\}$

stages can be improved by appropriately leveraging more explicit signals and monotonicity together.

4.5.1 Datasets

We consider four public datasets and contribute an additional large-scale dataset which contains various interaction types. These data cover different types of behavior chains and vary significantly in data sparsity.

- ***Steam*** [128]. This dataset covers a group of Australian users on the *Steam* video game distribution network[3] and was recently introduced for the task of bundle recommendation [128]. This dataset includes users' purchase information, play time of games they purchased, reviews, and thumbs-ups (i.e., 'recommended' or 'not recommended'). Based on these data we are able to build a 'purchase – play – review – recommend' chain for each user-item pair. Surprisingly, around 36% of purchased games are never played by users. Similarly the 'review' and 'recommend' actions are significantly sparser than the 'purchase' and 'play' actions (Table 4.1).
- ***YooChoose.***[4] This is a dataset provided by *YooChoose*[5] in the 2015 *RecSys Challenge*, which contains a series of click sessions and the purchase events that occurred in these sessions. Note that user IDs are not available in this dataset, thus we treat session IDs as user IDs and build 'click – purchase' chains for this data.
- ***Yelp.***[6] We use the Round 11 version of the *Yelp Challenge* data. We regard the reviews where rating scores are larger than 3 as 'recommend' actions and build 'review – recommend' behavior chains.
- ***GoogleLocal*** [64]. This dataset covers reviews about local businesses worldwide. It is a

[3] https://store.steampowered.com/
[4] http://2015.recsyschallenge.com/
[5] https://www.yoochoose.com/
[6] https://www.yelp.com/dataset/challenge

relatively sparse dataset, especially in terms of the number of interactions per user. We use the same criteria as the ***Yelp*** dataset to build 'review – recommend' chains.

- ***Goodreads.*** We introduce a new large-scale dataset from the book review website *Goodreads*.[7] This data contains 229,154,523 records collected from 876,145 users' public book shelves and covers 2,360,655 books (with detailed meta-data including authors, series, editions, publishers, numbers of pages, languages of book contents, similar books and top user-generated shelf names for these books). Each record contains information of a user's multiple interactions regarding an item, including date added to shelf, reading progress, rating score, and review text if available, thus making 'shelve – read – rate – recommend' chains available for our recommendation problem.

We apply the same preprocessing criteria for all five datasets: we discard users who have never reached the last stage of any behavior chain and items with fewer than 5 associated interactions in the system. Statistics and distributions of the above datasets after preprocessing are included in Table 4.1. For each dataset, we sample 100,000 interaction chains for validation and another 100,000 for testing. Within each of these two sets, each interaction chain corresponds to a different user. Data and code are available at `https://github.com/MengtingWan/chainRec`.

4.5.2 Comparison Methods and Evaluation Methodology

We consider three groups of methods for comparisons. We first consider methods where interaction stages are treated independently:

- **itemPop.** We count the observed interactions in the training set as preference scores for each stage. Thus items are ranked based on their popularity.
- **bprMF** [139]. Is a state-of-the-art pairwise ranking model for one-class recommendation. Independent latent factor models (Eq. (4.4)) are used for different interaction stages.
- **WRMF** [72, 123]. Is another line of models which optimizes the mean squared error

[7] `https://www.goodreads.com/`

between estimated preference scores and labels; additional weights are introduced to adjust unobserved interactions.

- **logMF** [80]. Similarly, independent latent factor models are applied here but the model optimizes a binary cross-entropy loss as in Eq. (4.2).

Next we consider alternative methods where the relationships among different interaction stages are involved.

- **condMF.** We adopt the conditional optimization criterion (Eq. (4.7)) and use independent latent factor models (Eq. (4.4)) to estimate the conditional probability $p_{ui,l|l-1}$.
- **condTF.** We apply the same optimization criterion but use tensor decomposition (Eq. (4.6)) to estimate the conditional probability.
- **sliceTF.** This is a combination of joint slicewise optimization (Eq. (4.5)) and tensor decomposition (Eq. (4.6)). Notice that this method can be regarded as extending the philosophy of existing work which jointly models different types of signals to our one-class problem on behavior chains [103, 124].
- **sliceTF (monotonic).** Uses the same learning strategy but replaces the original tensor decomposition by the monotonic scoring function (Eq. (4.11)).

Last we evaluate two implementations of the proposed algorithm—**chainRec (uniform)** and **chainRec (stagewise)**, where uniform sampling and stagewise sampling strategies are applied respectively.

By comparing methods from the first and the second groups, we evaluate whether incorporating multiple types of signals simultaneously can help with recommendation performance; by comparing the second and the last group of methods, we evaluate the effectiveness of the proposed techniques to leverage the special monotonic structure of these behavior chains.

We rank items based on the preference score $s_{ui,l}$ for each stage l, and consider the Area Under the ROC Curve (AUC) as an overall ranking metric, and Normalized Discounted Cumulative Gain (NDCG) as a top-biased evaluation metric. We fix the number of contrastive

samples for each positive user-item pair to $N = 1$ in all methods where such procedures are involved (e.g. Eq. (4.16))

4.5.3 Quantitative Results

We report detailed results with embedding dimensionality set to $K = 16$. We later perform a parameter study to assess sensitivity with respect to this parameter.

We include results for the primary item recommendation task in Table 4.2, where performance is evaluated based on users' feedback at the last stage. From this table we notice that the proposed **chainRec** algorithm significantly outperforms other baselines on most datasets in all metrics, though it performs slightly worse than **sliceTF** on ***Goodreads***. One notable advantage of **sliceTF** is that its plain linear structure makes it straightforward to optimize. Another possible reason for its good performance on ***Goodreads*** could be that it is a relatively dense dataset such that we have sufficient observations to implicitly learn the monotonicity property without needing to explicitly enforce it via model design. We observe that the second group of methods generally outperforms the first group of methods, which indicates that incorporating a spectrum of signals do help to predict the most sparse but explicit feedback. Surprisingly **sliceTF** consistently outperforms **sliceTF (monotonic)**. One possible reason could be that monotonic scoring functions behave as a kind of regularization on parameters and excessively and redundantly enforcing them may harm performance. Compared with this, the success of **chainRec** particularly validates the effectiveness of the proposed edgewise training strategy. Note that the non-personalized method **itemPop** performs as a strong baseline on the ***Steam*** dataset, possibly because the collected users' reviews are biased towards popular games as they were more likely to be exposed on the platform.

Next we evaluate item recommendation performance for each interaction stage separately. Results in terms of the AUC are included in Figure 4.5. For brevity we compare **chainRec** and two representative baselines, **logMF** and **sliceTF**, from the first and the second groups respectively.[8]

[8]Other baselines in general perform similarly to or weaker than these two methods.

Table 4.2: Results of the primary item recommendation task, which is evaluated based on users' most explicit feedback. The best performance is underlined and the last two columns show the percentage improvement of **chainRec** over the strongest baseline within each group.

		(a)				(b)				(c)			
Dataset	**Metric**	**itemPop**	**bprMF**	**WRMF**	**logMF**	**condMF**	**condTF**	**sliceTF**	**sliceTF (m.)**	**chainRec (uniform)**	**chainRec (stage.)**	**%impr. vs. (a)**	**%impr. vs. (b)**
Steam	AUC	0.955	0.963	0.963	0.962	0.961	0.959	0.967	0.957	0.964	0.968	0.44%	0.06%
	NDCG	0.318	0.318	0.314	0.319	0.298	0.310	0.278	0.266	0.319	0.323	1.21%	4.23%
YooChoose	AUC	0.914	0.924	0.920	0.922	0.929	0.920	0.940	0.928	0.951	0.950	2.90%	1.13%
	NDCG	0.140	0.152	0.154	0.150	0.124	0.133	0.185	0.154	0.199	0.176	28.73%	7.09%
Yelp	AUC	0.838	0.921	0.912	0.903	0.900	0.838	0.928	0.918	0.937	0.927	1.71%	0.91%
	NDCG	0.093	0.105	0.096	0.100	0.090	0.088	0.107	0.096	0.108	0.102	3.05%	0.60%
GoogleLocal	AUC	0.597	0.661	0.625	0.661	0.679	0.616	0.684	0.667	0.695	0.722	9.31%	5.69%
	NDCG	0.064	0.067	0.064	0.066	0.064	0.063	0.070	0.065	0.072	0.072	8.36%	2.92%
Goodreads	AUC	0.938	0.971	0.963	0.971	0.904	0.933	0.984	0.934	0.982	0.978	1.17%	-0.17%
	NDCG	0.124	0.125	0.098	0.127	0.072	0.104	0.132	0.121	0.132	0.113	3.94%	0.00%

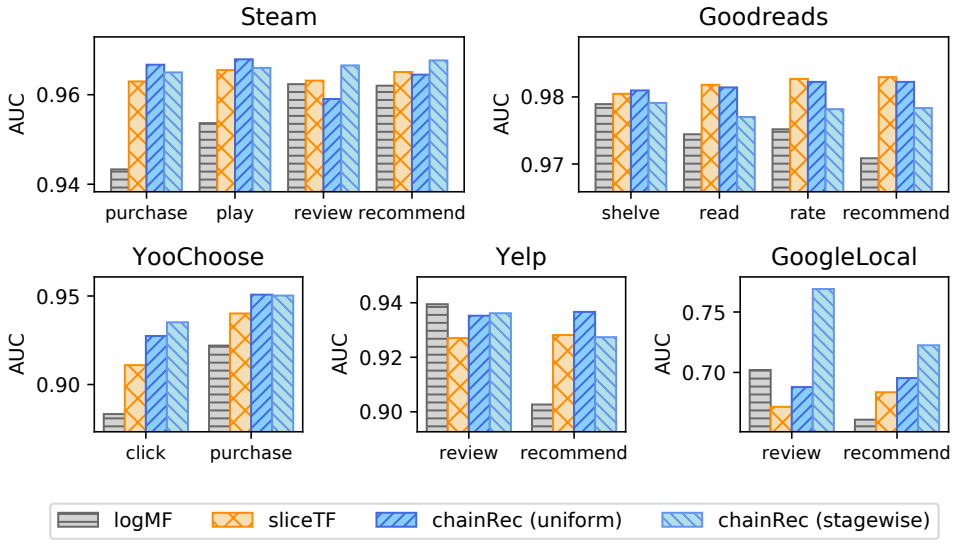


Figure 4.5: Results of item recommendation tasks on all stages in terms of AUC.

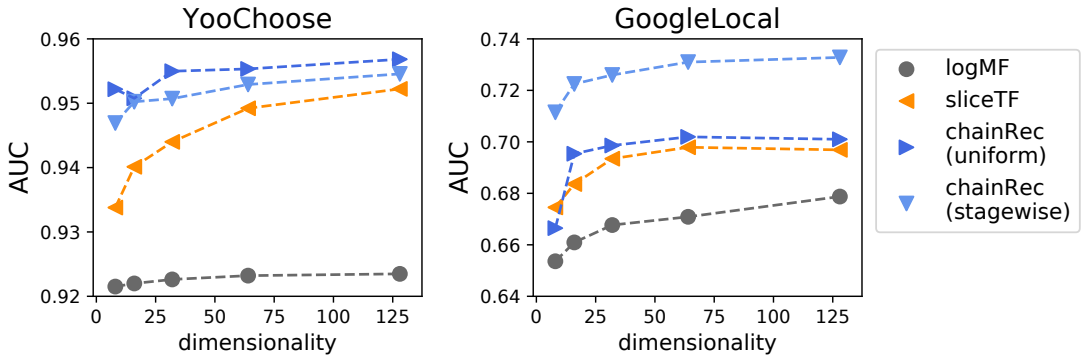


Figure 4.6: Sensitivity analysis w.r.t. dimensionality K on two datasets for the primary item recommendation task.

Here we find that **chainRec** and **sliceTF** yield better recommendation results on nearly all stages compared to training stage-specific standalone models (i.e., **logMF**), which implies that sufficiently exploiting users' interaction chains and appropriately leveraging monotonicity can help us to predict users' preferences across all stages generally.

We also vary the embedding dimension $K \in \{8, 16, 32, 64, 128\}$ and report the item recommendation performance at the last stage on ***YooChoose*** and ***GoogleLocal*** datasets in Figure 4.6, where **chainRec** still dominates other baselines as we increase the dimensionality.

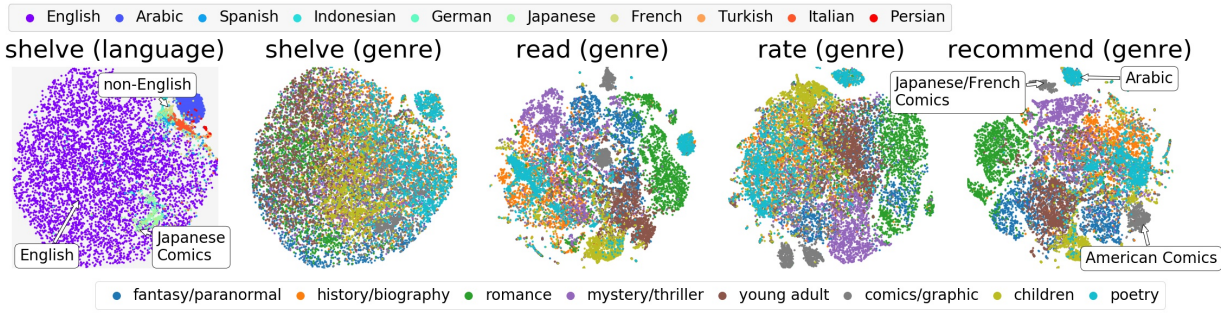


Figure 4.7: 2d t-SNE visualizations of item embeddings projected on different interaction stages (i.e., $\boldsymbol{\gamma}_i \circ \bar{\boldsymbol{\gamma}}_l$, where $\bar{\boldsymbol{\gamma}}_l = \boldsymbol{\gamma}_l / ||\boldsymbol{\gamma}_l||$ is the normalized stage-specific scalar). Different languages and genres of books are highlighted using different colors.

4.5.4 Qualitative Analysis

We apply the proposed **chainRec** algorithm on the ***Goodreads*** dataset and conduct case studies to explore the relationships among different stages.

We first seek to understand the stage-specific item embeddings learned from our model, by comparing these representations with the meta-data of items. We visualize the low-dimensional vectors of the normalized stage-specific item embeddings $\boldsymbol{\gamma}_i \circ \boldsymbol{\gamma}_l / ||\boldsymbol{\gamma}_l||$ in Figure 4.7 via t-SNE [106]. Here we categorize books into eight different genres: fantasy/paranormal, history/biography, romance, mystery/thriller, young adult, comics/graphic, children, and poetry. These genres are highlighted using different colors in Figure 4.7, from which we notice that these genre clusters are significantly visible for 'read' and 'recommend' actions but especially obscure for the 'shelve' action. This indicates that users tend to shelve books no matter what genres they belong to, but these features become important when users decide to read or recommend books.

We then investigate the 'shelve' action by highlighting the languages of book contents in Figure 4.7. Here we see a clear separation between non-English books and English books, with the only exception being a group of Japanese comic books that are mixed with English books. This observation indicates that language plays an important role when users shelve books. We further investigate the review texts associated with these Japanese comic books and find that some

English users might have mistakenly shelved Japanese editions despite the fact that what they read were English editions.

4.6 Conclusions and Future Work

In this study, we proposed an item recommendation framework to model the full spectrum of users' feedback. We observe that users' interactions often exhibit monotonic structure, i.e., the presence of a stronger (or more explicit) interaction necessarily implies the presence of a weaker (or more implicit) signal. After investigating alternative models, we proposed a new recommendation algorithm—**chainRec** which exploits all types of interactions and efficiently harnesses their monotonic dependencies. We contribute a new public dataset and validate the effectiveness of **chainRec** by quantitative and qualitative results on new and existing datasets.

We note that the monotonicity structures studied in this work are widely observable and a number of topics can be further explored along this trajectory. Beyond recommendation tasks, such monotonic dependency structures and the associated predictive models can potentially be extended to other areas such as (e.g.) medical diagnosis where dependencies exist between progressive symptoms. This monotonic chain structure and the proposed algorithm can also be extended to more general tree structures, where different branches (e.g. 'click–bookmark' and 'click–purchase–recommend' in e-commerce systems) can be modeled simultaneously. Empirically, we only consider the binary representation of each interaction stage but counts of interactions (e.g. play counts of music tracks) could be incorporated as confidences for these binary observations. We also plan to investigate more advanced sampling schemes, and further analysis of the edgewise optimization strategy.

4.7 Acknowledgements

This chapter contains the material as it appears in the *ACM Conference on Recommender Systems*, 2018 ("Item Recommendation on Monotonic Behavior Chains," Mengting Wan and Julian McAuley). The dissertation author was the primary investigator and author of this paper. We also thank Wang-Cheng Kang, Jianmo Ni and Shuai Tang for thoughtful discussions.

Chapter 5

Modeling Consumer Behavior with Unstructured Texts

5.1 Introduction

Consumers' unstructured textual feedback provides an incredible lens into the wide variety of opinions and experiences of different people, and play a critical role in helping users discover products that match their personal needs and preferences. In this chapter, we discuss two different frameworks to model consumer behavior with these unstructured texts: (1) recommending products by capturing consumers' action motivations buried in their *asymmetric* textual feedback; and (2) retrieving relevant information from consumers unstructured reviews to address complex and subjective product-related questions.

Personalized ranking with implicit feedback (e.g. purchases, views, check-ins) is an important paradigm in recommender systems. Such feedback sometimes comes with textual information (e.g. reviews, comments, tips), which could be a useful signal to reveal item properties, identify users' tastes and interpret their behavior. Although incorporating such information is common in *explicit* feedback settings (such as rating prediction), it is less common when dealing

with implicit feedback, as it is often not available for negative instances (e.g. there is no review associated with the item the user *didn't* buy). Thus in Section 5.2, we propose a ranking method (**PRAST**) to incorporate such personalized, asymmetric textual signals in implicit feedback settings.

Notice that in addition to standard product recommendation services, some review websites also allow users to pose product-related questions to the community via a question-answering (QA) system. As one would expect, just as opinions diverge among different reviewers, answers to such questions may also be subjective, opinionated, and divergent. This means that answering such questions automatically is quite different from traditional QA tasks, where it is assumed that a single 'correct' answer is available. Therefore, in Section 5.3, we introduce the idea of question-answering using unstructured review texts, with the emphasis on the following two aspects: (1) questions have multiple, often divergent, answers, and this full spectrum of answers should somehow be used to train the system; and (2) what makes a 'good' answer depends on the asker and the answerer, and these factors should be incorporated in order for the system to be more personalized.

5.2 Recommending Products with Asymmetric Textual Feedback

Textual information associated with user-item pairs (e.g. review text) has proven helpful when explaining and predicting explicit feedback (e.g. rating prediction), particularly on 'cold' items [18, 43, 101, 111, 163]. The principle of these approaches relies on factorizing observed ratings and modeling review text by linking latent preference dimensions and topics discovered in text. The success of these methods motivates us to adopt textual information in implicit feedback settings. Specifically, rather than uncovering 'facets' from review text that explain users' ratings, we would like to use these textual signals to learn about the *types* of actions users are likely to

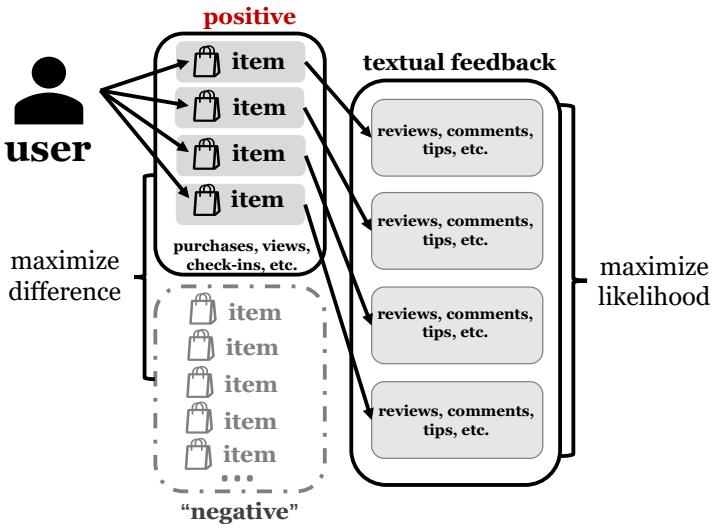


Figure 5.1: Illustration of asymmetric textual information in implicit feedback settings.

perform. For example, we might wish to uncover the aspects of an item from Amazon reviews, or Youtube comments, which may trigger a 'purchase' or 'view' action. However, in addition to the label asymmetry described above, this textual information is also asymmetric. This means that unlike in explicit feedback settings where all responses used for training have the same side-information, this is no longer the case in implicit-feedback settings, where such textual information is only available for positive user-item pairs. We address this asymmetry and describe our goal in this paper as follows:

- **Goal:** *Given (asymmetric) textual information, we seek to understand users' inclinations towards particular kinds of actions, and provide item recommendations guided by these signals.*

In order to incorporate such textual information and overcome the challenge of asymmetry, we propose a new one-class recommendation model – ***P**airwise **R**anking with **A**symmetric **T**extual Feedback* (**PRAST**), where we assume: (1) positive and negative items differ in terms of their compatibility with a given user, which is consistent with the typical pairwise ranking optimization criterion (e.g. **BPR**); (2) relevant asymmetric textual data are consistent among observed positive

items, such that they provide information related to the likelihood of users' actions.

The above suggests some form of joint objective, where our model of users and items should be good at predicting (or 'explaining') observed versus non-interactions, but should also be good at explaining (in terms of likelihood or perplexity) the textual information associated with positive actions. We apply the proposed framework on two large-scale datasets and show that item ranking performance can be significantly improved by appropriately incorporating asymmetric textual data. Our experiments reveal that such side-information not only helps to provide better recommendations, but also can be used to uncover the motivations behind observed user-item interactions.

In conclusion, our goal in this study is to propose a ranking method to incorporate such personalized, asymmetric textual signals in implicit feedback settings. We evaluate our model on two real-world datasets. Quantitative and qualitative results indicate that the proposed approach significantly outperforms standard recommendation baselines, alleviates 'cold start' issues, and is able to provide potential textual interpretations for latent feedback dimensions.

5.2.1 Related Work

Traditional models for item recommendation rely on techniques such as Collaborative Filtering (CF) to learn from explicit feedback like star-ratings [89]. Although several paradigms for explicit feedback exist, of most relevance to us are *model-based* methods and in particular Matrix Factorization (MF) methods [90]. Such models have been extended in order to handle implicit feedback data where only positive signals (e.g. purchases, views, clicks) are observed (i.e., the so-called 'one-class' recommendation setting). Most relevant here are pair-wise methods like **BPR-MF** [139] that make an assumption that positive feedback instances are simply 'more preferable' than non-observed feedback.

Several models exist that incorporate textual feedback to predict star ratings, including **HFT** ('Hidden Factors and Topics') [111], **JMARS** ('Jointly Modeling Aspects, Ratings, and

Sentiments') [43], **RMR** ('Ratings Meet Reviews') [101], **FLAME** ('Factorized Latent Aspect Model') [163] and **SLUM** ('Sentiment Utility Logistic Model') [18]. These models differ from each other in precise formulation, but each essentially assumes that reviews can be used to determine the 'aspects' along which users rate products, using fewer observations than would be required to learn these aspects from ratings alone. This is a natural assumption, as the very purpose of reviews is to explain the different factors that contributed to a user's rating. We rely on a similar assumption, though the 'aspects' we seek to discover should discriminate interactions from non-interactions (e.g. purchases from non-purchases, views from non-views), and thus are quite different.

Similar to the problem we tackle, several works have attempted to incorporate side-information into implicit feedback settings and have proven helpful when handling 'cold-start' issues. Examples include extensions of BPR, such as Social BPR (**SBPR**), which makes use of side information in the form of social signals [177], where friends' activities act as a form of implicit signal that guides users' actions; and Visual BPR (**VBPR**), where visual attributes are used to estimate item 'facets' that guide users' purchases [66]. In particular, some studies have been proposed to incorporate item-associated textual content (e.g. the content of an article) into this setting where topics in text are used to guide latent item dimensions [155, 167]. Although such information (social networks, images, article texts) have been shown to be effective in such cases, this is different from the setting we study as it does not exhibit the same *asymmetry*: the feedback in question is 'static' (images and article texts are used to extract item features, social networks are used to extract user features), and depends on the user or the item only, not the user-item *interaction*.

5.2.2 Methodology

In order to gradually construct our new framework—Pairwise Ranking with Asymmetric Textual Feedback (**PRAST**), we first formally introduce the traditional latent factor model and the

Bayesian Personalized Ranking (**BPR**) framework as background information. Suppose $>_u$ is the desired preference ranking for user u, and I_u^+ and I_u^- are the positive item set and the unobserved (or 'negative') item set. Then our training data for ranking based on implicit feedback consists of a sequence of (*user*, *positive-item*, *negative-item*) triples, i.e.,

$$D_S = \{(u,i,i') \mid u \in U \wedge i \in I_u^+ \wedge i' \in I_u^-\}. \tag{5.1}$$

In the **BPR** framework [139], the following ranking-based likelihood is optimized:

$$\prod_{u \in U} P(>_u |\Omega) = \prod_{(u,i,i') \in D_S} P(i >_u i' |\Omega), \tag{5.2}$$

where Ω is the parameter set. Here $i >_u i'$ indicates that user u prefers item i over item i' and its probability is usually defined via a sigmoid function:

$$P(i >_u i' |\Omega) = \sigma(s_{u,i} - s_{u,i'}) = \frac{1}{1 + e^{-(s_{u,i} - s_{u,i'})}},$$

where the latent factor model Eq. (2.1) can be applied for the preference score.

We present a new one-class recommendation model—Pairwise Ranking with Asymmetric Textual Feedback (**PRAST**), where an enhanced pairwise ranking optimization criterion is applied to handle evidence such as (asymmetric) textual information, and a relevance-aware topic model is attached to the latent factor model so that text can be incorporated adaptively.

Overview of the Framework.

In order to construct a ranking framework with asymmetric textual information, we consider the likelihood of the desired rankings as well as the 'appearance probability' of the observed 'positive-only' text. Then we consider the following training data which consists of a

set of (*user*, *positive-item*, *negative-item*, *evidence*) quadruples, i.e.,

$$D_S = \{(u, i, i', E_{u,i}) \mid u \in U \wedge i \in I_u^+ \wedge i' \in I_u^-\}. \tag{5.3}$$

Here the evidence $E_{u,i}$ could be represented either via the 'positive-only' text corpus $R_{u,i}$, or empty (i.e. no textual information associated with the observed action). Then we wish to maximize (the logarithm of) the following likelihood:

$$\prod_{(u,i,i',E_{u,i}) \in D_S} \underbrace{P_\Omega(E_{u,i}|i >_u i')}_{\text{likelihood of the evidence}} \underbrace{P_\Omega(i >_u i')}_{\text{pairwise ranking}}.$$

We use P_Ω as shorthand to denote the probability given the parameter set Ω. The latent factor model Eq. (2.1) and the sigmoid transformation in **BPR** Eq. (5.2) can be applied to model $P_\Omega(i >_u i')$ as well. Thus we naturally inherit the optimization principle from **BPR**: compatibilities between positive and negative instances can be fairly compared through latent factors and the difference can be maximized. If there is textual information associated with the triple (u, i, i') (e.g. a review was left after user u purchased a product i), we define the likelihood of asymmetric evidence $E_{u,i}$ as a monotonic function of the likelihood of the interaction-associated text document $R_{u,i}$, i.e.,

$$P_\Omega(E_{u,i}|i >_u i') = P_\Omega(R_{u,i})^\kappa. \tag{5.4}$$

Here κ is a positive hyperparameter which is used to control the confidence of the underlying language model for $R_{u,i}$. In order to use such textual information to explain and facilitate pairwise ranking, we need to fuse latent dimensions in Eq. (2.1) with the language model. Thus in Eq. (5.4), larger κ indicates higher confidence that observed textual data are bonded to motivations of the target action. As $\kappa \to 0$, $P_\Omega(R_{u,i})^\kappa \to 1$ for all $R_{u,i}$, which implies textual data are ignored and only the pairwise ranking is considered during the training process. Specifically, we define $P_\Omega(E_{u,i,i'}|i >_u i') = 1$ if there is no textual information provided. Because of the asymmetry of

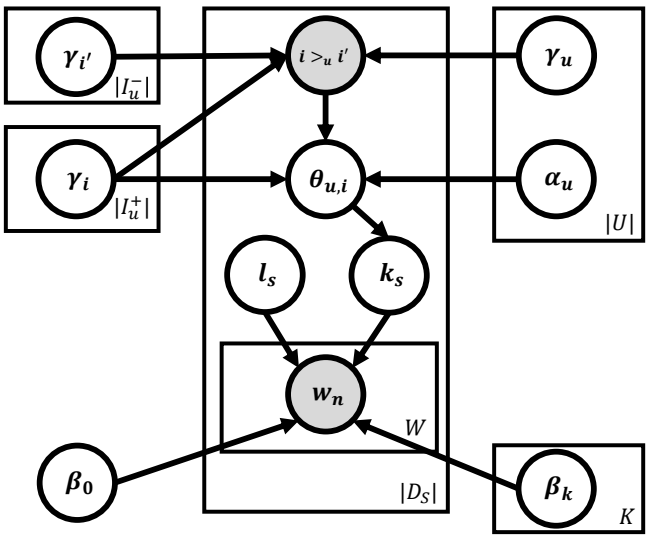


Figure 5.2: Plate-notation illustration of the proposed **PRAST** model.

$R_{u,i}$, we always assume there is no additional information at test time and the predictions we can provide are purely a function of the user and item representations, i.e., the preference scores $s_{u,i}$ as determined by the latent factor model Eq. (2.1).

Language Model.

Topic models have proven a popular approach to incorporate textual information into latent factor models, by combining Latent Dirichlet Allocation (LDA) [22] with latent factor models [43, 101, 111, 163]. Just as LDA uncovers hidden dimensions in documents, when combined with a latent factor model it can uncover those dimensions that explain variance in people's opinions as represented by rating scores. Based on a similar principle, in implicit feedback settings, we consider distinguishing whether a sentence is relevant to the target behavior and only attach the relevant part to the latent preference dimensions, so that these relevant contents can be consistently and adaptively explained among positive items while others might be explained by a background model. We gradually build the language model as follows.

- **Sentence Relevance.** For each sentence s in a document $R_{u,i}$, we introduce another binary

latent variable l_s to model sentence relevance.[1] We assume that $P_\Omega(l_s = 1) = P_\Omega(l_s = 0) = 0.5$. Then the corpus likelihood in Eq. (5.4) can be modeled as

$$P_\Omega(R_{u,i}) = C \prod_s \Big(l_s P_\Omega^{(1)}(R_{u,i,s}) + (1 - l_s) P_\Omega^{(0)}(R_{u,i,s}) \Big),$$

where C is a constant and $P_\Omega^{(l)}(R_{u,i,s})$ is shorthand for $P_\Omega(R_{u,i,s}|l_s = l)$.

- **Topic Distribution.** For the relevant textual contents (i.e. $l_s = 1$), similar to **LDA**, we have a K-dimensional topic distribution $\boldsymbol{\theta}_{u,i}$ for each document, which indicates the probability that a particular word in this document discusses a particular topic. We apply the item latent factor $\boldsymbol{\gamma}_i$ in Eq. (2.1) and introduce another user-specific non-negative K-dimensional parameter $\boldsymbol{\alpha}_u$ to model this distribution as follows:

$$\theta_{u,i,k} = \frac{\exp(\alpha_{u,k}\gamma_{i,k})}{\sum_{k'=1}^{K} \exp(\alpha_{u,k'}\gamma_{i,k'})}, \tag{5.5}$$

  where $\alpha_{u,k} \geq 0, \forall k$. Recall that $\boldsymbol{\gamma}_i$ captures item i's 'properties' in the latent factor model, and here we use $\boldsymbol{\alpha}_u$ to capture variation due to user u's writing style (i.e., a user-specific topic weighting determining which topics this user prefers to write about in their reviews). By applying such a transformation, we are assuming that if an item exhibits certain properties which may motivate users to take actions (i.e., high $\gamma_{i,k}$), then these aspects should be reflected in users' reviews, so long as that user has a tendency to discuss them (high $\alpha_{u,k}$). This modeling approach is an enhanced version of that of the **HFT** model [111] proposed for explicit-feedback settings, where $\alpha_{u,k}$ is assumed to be constant among all users and all topics.

- **Topic Assignment.** Furthermore, we assume that words within a sentence s discuss the same aspect k_s (i.e., topic) of the item. Such a sentence-level topic is generated from a multinomial distribution with its corresponding review topic parameter $\boldsymbol{\theta}_{u,i}$. Note here the

[1] $l_s = 1$ indicates this sentence is relevant to motivations of the observed user-item interaction; $l_s = 0$ otherwise.

each sentence-level topic k_s is a latent variable in the probabilistic model, which is usually estimated through sampling approaches [133] or variational inference [151].

- **Word Distribution.** Suppose $\mathscr{W}$ is the dictionary used in the model, $\boldsymbol{\beta}_0, \boldsymbol{\beta}_k, k = 1, \ldots, K$ are W-dimensional vectors where W is the dictionary size. Then given the sentence relevance and topic assignment, we could generate the complete review text $R_{u,i}$ from word distributions. Specifically, for a relevant sentence, given the topic assignment k_s for a word w_n, its likelihood can be modeled as

$$\phi_{w_n,k_s} \coloneqq P_\Omega(w_n|k_s, l_s = 1) = \frac{\exp(\beta_{k_s,w_n})}{\sum_{w' \in \mathscr{W}} \exp(\beta_{k_s,w'})}.$$

  For an irrelevant sentence, we have the following background word distribution:

$$\phi_{w_n,0} \coloneqq P_\Omega(w_n|l_s = 0) = \frac{\exp(\beta_{0,w_n})}{\sum_{w' \in \mathscr{W}} \exp(\beta_{0,w'})}.$$

Therefore, the final likelihood of textual information in each sentence is

$$P_\Omega(R_{u,i,s}) = l_s\big(\sum_{k_s} \theta_{u,i,k_s} \prod_w \phi_{w,k_s}\big) + (1 - l_s)\big(\prod_w \phi_{w,0}\big).$$

The graphical representation of the complete model is included in Figure 5.2.

Model Inference.

We apply an EM-style variational inference method to fit the text term $P_\Omega(R_{i,u}|i >_u i')$ and the ranking term $P_\Omega(i >_u i')$ jointly, which is similar to the techniques applied in previous textual model studies [46, 163]. To do so we introduce an intermediate parameter τ_s for sentence relevance indicator l_s, which can be easily updated in the E-step: $\tau_s = P_\Omega^{(1)}(R_{u,i,s})/(P_\Omega^{(1)}(R_{u,i,s}) +$

$P_\Omega^{(0)}(R_{u,i,s})$). Then our target is to maximize the following log-likelihood for each sentence:

$$\tau_s \log P_\Omega^{(1)}(R_{u,i,s}) + (1-\tau_s)\log P_\Omega^{(0)}(R_{u,i,s}) \tag{5.6}$$

Then we introduce another set of K-dimensional variational parameters $\boldsymbol{\pi}_s$ to approximate the distribution of the sentence topic assignment k_s, i.e., the variational probability is $q(k_s = k|\boldsymbol{\pi}_s) = \pi_{s,k}$, with the constraint $\sum_k \pi_{s,k} = 1$. Instead of optimizing the original text-related log-likelihood $\log P_\Omega^{(1)}(R_{u,i,s})$, we maximize the lower-bound of this log-likelihood as

$$\begin{aligned}
&\mathbb{E}_q \log P_\Omega^{(1)}(R_{u,i,s}) - \mathbb{E}_q \log q(k_s|\boldsymbol{\pi}_s) \\
=&\sum_k \pi_{s,k}\Big(\log\theta_{u,i,k} + \sum_{w_n \in R_{u,i,s}} \log\phi_{w_n,k} - \log\pi_{s,k}\Big) \\
=&\sum_k \pi_{s,k}\Big(\log\theta_{u,i,k} + \sum_{w\in\mathscr{W}} \log N_{s,w}\phi_{w,k} - \log\pi_{s,k}\Big),
\end{aligned}$$

where $N_{s,w}$ is the frequency of word w in sentence s. In practice, we first fix all other parameters and update τ_s and $\pi_{s,k} \propto \theta_{u,i,k}\prod_w \phi_{w,k}^{N_{s,w}}$. Then we fix $\tau_s, \pi_{s,k}$ and update other parameters to maximize the above lower-bound plus the log-likelihood of the background language model $\log P_\Omega^{(0)}(R_{u,i,s})$ and the pairwise ranking $\log P_\Omega(i >_u i')$.

Gaussian priors are included for all parameters in Ω, leading to a standard ℓ_2 regularizer. In addition, we apply the **ADAM** optimizer [84], a stochastic gradient-based algorithm. Recall that our primary goal is to produce rankings that are consistent with our training data (i.e., positive instances should be ranked highly). Thus we need to be careful not to overfit too much to side information, which would sacrifice ranking quality. Rather, the textual information is intended to regularize or 'reinforce' the model's latent factors, in order to lead to better ranking performance. Therefore, during stochastic optimization, we periodically compute the ranking measure (i.e., the AUC) on a held-out validation set. We report results on the test set for the model parameters, hyperparameters, and the iteration, leading to the best performance on the validation set.

5.2.3 Experiments

We evaluate the proposed **PRAST** model for personalized item ranking on two large-scale datasets where asymmetric textual feedback is available. In particular, we evaluate (1) whether overall item rankings can be estimated more accurately by leveraging such signals; (2) whether *cold start* issues for items can be alleviated; (3) whether latent preference dimensions can be reasonably explained by textual information and motivation-relevant topics can be discovered.

Datasets

We consider two large-scale datasets—***Amazon*** [112] and ***Google Local*** [64], where both review text and ratings are available. Recall that we do not use rating information (instead we are trying to predict what items a user would 'interact' with, such as what business they would visit), except when adopting explicit-feedback models for comparison.

- **Amazon.** This is a large-scale dataset collected from *Amazon.com* [112]. We consider products in eight top-level categories: ***Instant Video***, ***Office Products***, ***Digital Music***, ***Baby***, ***Pet Supplies***, ***Grocery and Gourmet Food***, ***Health and Personal Care*** and ***Cell Phones and Accessories***. We discard users with fewer than 3 associated actions (i.e., reviews) in total leaving around 4 million actions across 807 thousand items and 849 thousand users. Textual information is available for almost all actions. Models are built independently for different categories and the per-category statistics are included in Table 5.1.
- **Google Local.** The ***Google Local*** dataset was introduced in a recent paper [64], which contains reviews about local businesses worldwide. We extract businesses from the following states in the US: ***Colorado***, ***North Carolina***, ***Washington***, ***Illinois***, ***Florida***, ***New York***, ***Texas*** and ***California***. Similarly we discard users with fewer than 3 actions and build models independently for different states. This results in around 1 million actions across 518 thousand items and 184 thousand users, around 71% of which have associated textual information. Compared with ***Amazon***, ***Google Local*** is a relatively sparse dataset in terms

of actions associated with items, and contains relatively shorter reviews.

Intuitively we consider 'review' actions as positive feedback in our experiments, i.e., we regard all of the reviewed user-item pairs as positive. Appearance of this action indicates that a user bought a product or visited a place. Different forms of implicit feedback could be considered (such as clicks or purchases, if such data were available), but using the presence of reviews is desirable as it allows us to straightforwardly compare against models designed for explicit feedback settings, as described below.

Baselines and Evaluation Methodology

We consider the following implicit-feedback baselines:

- **itemPop.** As item popularity (i.e., the number of previous actions regarding an item) could be a significant component in item ranking, we simply use the count of positive responses for each item in the training set as its preference score so that items are ranked in terms of their popularity.
- **BPR.** This is a state-of-the-art implicit-feedback pairwise ranking model. As we introduced previously, a latent factor model is applied to generate item preference scores.
- **WARP.** Weighted Approximate-Rank Pairwise [161] is another state-of-the-art loss for Top-K recommendation, which penalizes positive items at lower rank heavily. Specifically, we apply a penalizing scheme similar to [70], where a positive item i based on its rank $w_{i,u} = \log(rank_{i,u} + 1)$.
- **WRMF.** Weighted Regularized Matrix Factorization [72, 123] is another family of implicit-feedback models, where standard matrix factorization is applied and an additional weight is introduced to model unobserved interactions, i.e., the loss function takes the form $\sum_{u,i} c_{u,i}(y_{u,i} - s_{u,i})^2$, where $y_{u,i} \in \{0,1\}$ is the label of the feedback and $c_{u,i}$ is usually set to be large for positive feedback but small for non-interactions.

Comparing these implicit-feedback methods against **PRAST** allows us to measure the influence

of (asymmetric) textual feedback in terms of ranking quality.

In addition, we consider two more alternatives that make use of the same textual information: (1) a representative probabilistic model from a series of methods where review text is incorporated into rating prediction, and (2) a state-of-the-art model designed for point-of-interest (POI) recommendation where 'tip' texts are included in order to estimate the number of users' check-ins. In particular we consider the following two models:

- **HFT-b.** Hidden Factors as Topics (**HFT**) [111] is an explicit-feedback approach which models both review text and ratings. We still consider the observed reviews only and replace the original Meas Squared Error (MSE) loss by a binary cross-entropy loss: $y_{u,i} \log \sigma(s_{u,i}) + (1 - y_{u,i}) \log(1 - \sigma(s_{u,i}))$, where $y_{u,i} = 1$ if the rating score is larger or equal to 3 and $y_{u,i} = 0$ otherwise.
- **CAPRF-b.** The Context-Aware POI Recommendation Framework **CAPRF** [52] applies a similar loss function to **WRMF** but the number of check-ins is regarded as a label $y_{u,i}$ and all non-interactions are discarded. Here, tip texts are modeled as an additional regularization of item- and user- latent factors through linear embeddings. In our case, similar to **HFT**, we replace the number of check-ins by a binary label based on rating score.

Note that the above two baselines use the same textual information as our method but discard all non-interactions; thus they require minor adaptation to apply them in our implicit-feedback setting: The basic assumption behind our adaptation of these methods is that users are likely to interact with (purchase, visit, or consume) items for which they are predicted to exhibit a high preference score, based on their explicit signals (e.g. high rating scores, multiple check-ins). By comparing these two methods against **PRAST**, we address the difference between explicit-feedback and implicit-feedback objectives and evaluate the influence of taking abundant unobserved interactions into consideration given the same amount of textual information.

As our goal is to provide high-quality personalized item rankings, we adopt the Area Under the ROC Curve (AUC) as the overall evaluation measure (which is also the criterion

that **BPR** variants optimize), as well as Normalized Discounted Cumulative Gain (NDCG) as a top-biased ranking measure.

Quantitative Results

Following [111] we set $K = 10$ for the dimensionality of latent factor vectors and the number of topics. The confidence parameter for the language model κ is set to be 0.1 in all the experiments and the regularization parameter $\lambda \in \{0.01, 0.05, 0.1, 1\}$ is selected based on validation performance. We apply leave-one-out evaluation, where for each dataset, we sample 5000 users and their last action for testing, and their second-to-last action for validation. All other actions in the dataset are used for training. All results are reported on the held-out test data.

We include the overall results in terms of the AUC and NDCG on ***Amazon*** and ***Google Local*** datasets in Table 5.2. To address the 'cold-start' problem, in addition to the complete dataset, we report the performance on 'cold' items where the number of associated actions is less than 5. For brevity, we include only the average AUC/NDCG (across all categories and states) for ***Amazon*** and ***Google Local***, and provide barplots of AUC (on the complete dataset) for each product category and each state in Figure 5.3.

From Table 5.2 we notice that **PRAST** significantly outperforms standard implicit-feedback baselines and adjusted explicit-feedback baselines in terms of overall ranking (AUC), especially when recommending 'cold' items. This indicates that appropriately incorporating textual information into a ranking loss can improve personalized item recommendations. Based on Figure 5.3 and Table 5.1, this improvement is substantial on 'sparse' datasets in terms of the number of actions per item (e.g. ***Amazon (Digital Music)***) but less significant on relatively 'dense' datasets (e.g. ***Amazon (Instant Video)***). For ***Google Local***, we observe limited impact of textual information on small datasets (e.g. ***Colorado***). One possible reason for this could be the lack of data to fit high-dimensional language models.

For top-biased evaluation (NDCG), **PRAST** outperforms baselines on both ***Amazon***

Table 5.1: Basic dataset statistics: numbers of actions (i.e. reviews), users, items, sentences, actions per item, sentences per document.

Amazon	**#act.**	**#users**	**#items**	**#act. /item**	**#sent.**	**#sent. /act.**	**Google Local**	**#act.**	**#users**	**#items**	**#act. /item**	**#sent.**	**#sent. /act.**
Instant Video	135K	29,756	15,149	8.92	608K	4.50	Colorado	72K	10,512	27,984	2.57	233K	3.24
Office Prod.	287K	59,858	60,641	4.73	1,540K	5.37	North Carolina	73K	10,644	33,071	2.21	214K	2.93
Digital Music	352K	56,814	156,503	2.25	1,821K	5.18	Washington	78K	9,699	29,644	2.63	194K	2.49
Baby	380K	71,826	42,523	8.94	2,071K	5.45	Illinois	135K	17,098	42,329	3.18	377K	2.80
Pet Supplies	478K	93,336	70,105	6.82	2,331K	4.88	Florida	182K	28,898	81,205	2.24	539K	2.96
Grocery	509K	86,400	108,467	4.69	2,390K	4.70	New York	225K	22,199	61,790	3.65	579K	2.57
Health	1,073K	205,704	163,717	6.56	5,126K	4.78	Texas	266K	35,547	96,597	2.75	761K	2.86
Cell Phones	1,079K	245,110	190,089	5.67	4,664K	4.32	California	430K	48,957	145,779	2.95	982K	2.28
Total	4,293K	848,804	807,194	5.32	20,550K	4.79	Total	1,461K	183,554	518,399	2.82	3,879K	2.66

Table 5.2: Results on ***Amazon*** and ***Google Local*** (average metric across the complete dataset). The best performance is underlined and the last column shows the percentage improvement of **PRAST** over the strongest baseline.

Dataset	**Metric**	**itemPop**	**BPR**	**WARP**	**WRMF**	**HFT-b**	**CAPRF-b**	**PRAST**	improv. vs. BPR	improv. vs. HFT	improv. vs. best
Amazon (overall)	AUC	0.7806	0.7990	0.7917	0.7881	0.7724	0.7846	0.8194	2.55%	6.09%	2.55%
	NDCG	0.1079	0.1052	0.1071	0.1077	0.1010	0.0984	0.1082	2.88%	7.15%	0.29%
Amazon (cold)	AUC	0.5675	0.6000	0.5869	0.5804	0.5588	0.5691	0.6364	6.06%	13.88%	6.06%
	NDCG	0.0713	0.0711	0.0712	0.0713	0.0705	0.0706	0.0715	0.49%	1.32%	0.28%
Google Local (overall)	AUC	0.5458	0.6731	0.5932	0.5730	0.5718	0.5751	0.7068	5.00%	23.60%	5.00%
	NDCG	0.0809	0.0786	0.0766	0.0805	0.0825	0.0813	0.0869	10.51%	5.29%	5.29%
Google Local (cold)	AUC	0.5043	0.6459	0.5625	0.5339	0.5323	0.5346	0.6798	5.25%	27.70%	5.25%
	NDCG	0.0752	0.0761	0.0735	0.0746	0.0762	0.0747	0.0804	5.59%	5.49%	5.49%

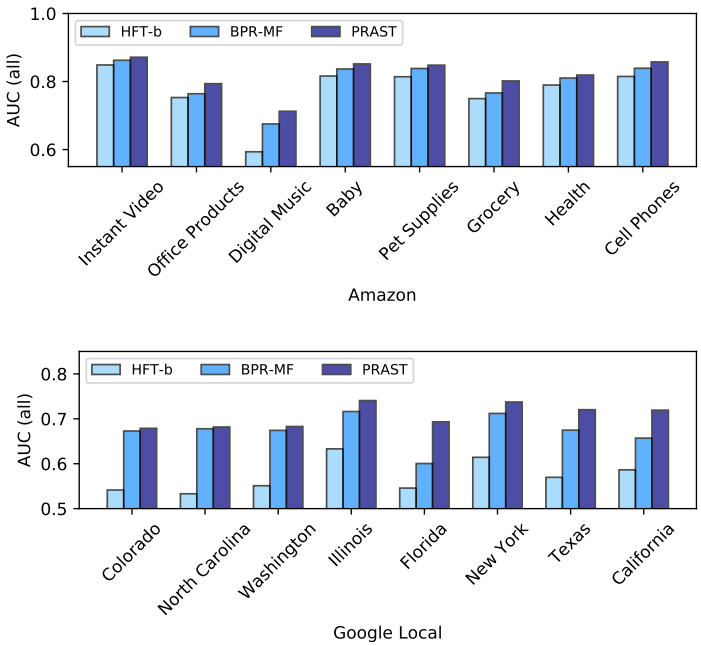


Figure 5.3: Results for each category in ***Amazon*** and for each state in ***Google Local*** in terms of the AUC.

and ***Google Local*** datasets, though the improvement is relatively limited on ***Amazon***. This is possibly because the number of items is very large and item popularity often dominates user preferences (especially on some ***Amazon*** categories) so that improving a top-biased ranking metric is relatively difficult for 'cold' items.

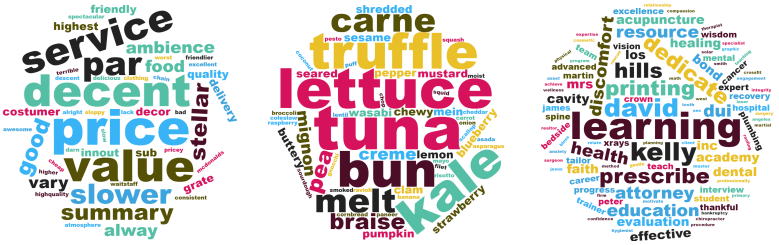
Qualitative Analysis

We examine the top words for the topics discovered from **PRAST** based on the normalized topic-specific word likelihood $\frac{\phi_{w,k}}{\sum_{k'} \phi_{w,k'}}$. Such topics can be used to explain latent preference/motivation dimensions. Word clouds of three topics discussed in ***Amazon (Office Products)*** and ***Google Local (California)*** are shown in Figure 5.4. In general, topics uncovered from ***Amazon (Office Products)*** are a mixture of genres (e.g. printer/scanner in Figure 5.4a) and aspects (e.g. price/shipping in Figure 5.4b or product appearance Figure 5.4c), which reveal what product a user wants to buy and what aspect a user cares about when making a purchase. Similarly, we

Amazon (Office Products)

(a) Printer/Scanner (b) Price/Shipping (c) Appearance

Google Local (California)



(d) Restaurant(General) (e) Restaurant(Food) (f) Services

Figure 5.4: Word clouds from three selected topics addressed in textual feedback from ***Amazon (Office Products)*** and ***Google Local (California)***.

see categories (e.g. education/medical/law services in Figure 5.4f) from topics in ***Google Local (California)*** and even different viewpoints for a particular category (e.g. general aspects like price and service for restaurants in Figure 5.4d, and particular foods in Figure 5.4e).

In Figure 5.5, we provide an example review for an item with relatively high scores for the item latent factors in $\boldsymbol{\gamma}_i$ associated with the topic 'print/scanner' and the topic 'price/shipping'. We split the review text into sentences and provide the estimated relevance score τ_s for each sentence in parentheses. We notice that **PRAST** can automatically highlight the sentences containing genre-specific keywords (e.g. ink, printer) and price-related contents (e.g. cost, price, money, economical) by assigning high relevance scores. This suggests that in addition to improving ranking quality, the **PRAST** model is capable of explaining latent feedback dimensions and detecting the most relevant textual content.

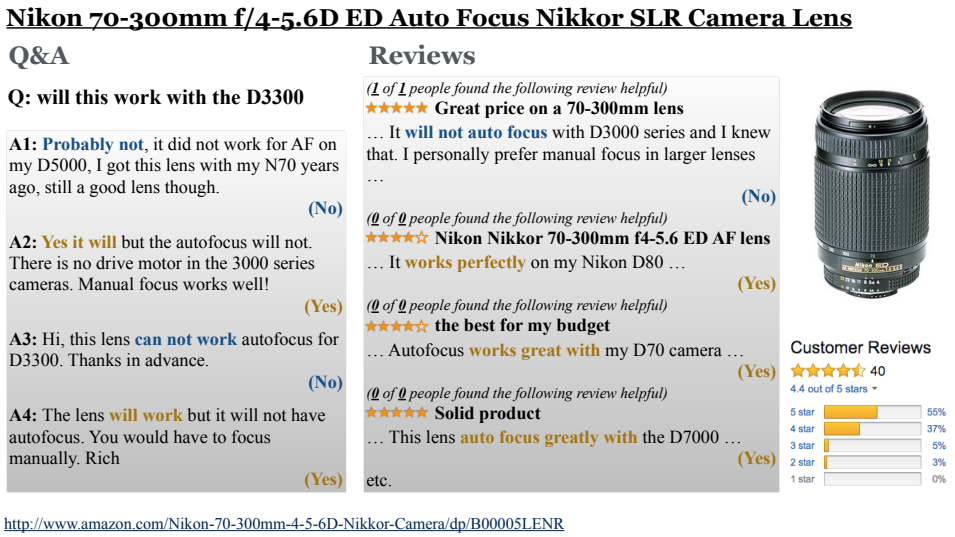
Epson WorkForce 840 Wireless All-in-One Color Inkjet Printer, Copier, Scanner, Fax

1) I have used it for several months now - Like it a lot - it has been reliable and easy to install. *(0.321)*
2) For me the most important factor is cost of use. (0.540)
3) Is it economical. (0.490)
4) It is economical and one very nice feature is it uses pigment ink so it is dry upon printing. (0.968)
5) It is hard to give any printer 5 stars as these are very good reasonably priced items but they are not expensive state of the art printers. (0.985)
6) One is clearly getting a lot for one's money. (0.761)

Figure 5.5: An example review selected from an item with large scores on the 'printer/scanner' and the 'price/shipping' dimensions, where the estimated sentence relevance scores τ_s are provided in parentheses.

5.3 Addressing Complex Product-Related Queries with Textual Consumer Feedback

Users' unstructured textual feedback (e.g. review texts) are a valuable resource to help people make decisions. This kind of feedback data may contain a wide range of both objective and subjective product-related information, including features of the product, evaluations of its positive and negative attributes, and various personal experiences and niche use-cases. In addition to passively searching for information that users are interested in among reviews, a number of e-commerce websites, such as *Amazon* and *ebay*, also provide community question answering systems where users can ask and answer specific product-related questions. While such systems allow users to seek targeted information (as opposed for searching for it in reviews), asking the community is still time-consuming in the sense that the user must wait for a response, and even then may have quite different preferences from the user who answers their questions. The above issues motivate us to study systems that help users to automatically navigate large volumes of



http://www.amazon.com/Nikon-70-300mm-4-5-6D-Nikkor-Camera/dp/B00005LENR

Figure 5.6: A real opinion QA example from *Amazon.com*. The left box shows answers provided by the community, demonstrating the divergent range of responses. The right box shows the type of system we develop to address such questions, mining divergent and subjective opinion information from product reviews.

reviews in order to locate relevant and informative opinions, in response to a particular query.

This kind of 'opinion question answering' system (opinion QA) is quite different from typical community question answering (cQA) systems. In particular, traditional cQA systems are usually concerned with *objective* information, such that answers can be generated by constructing and exploring a knowledge-base which is composed of facts. However, for the opinion QA problem, users often ask for *subjective* information, such as "Is this a good lens for my Nikon D3300 camera?" Such a seemingly simple question is complex because it depends on (a) objective information (is the lens even compatible?); (b) subjective information (whether it's 'good' is a matter of opinion); and (c) personalization (which answer is correct *for the user asking the question*; are they an expert? an amateur? on a budget? etc.). Perhaps not surprisingly, opinion QA systems generate a wide variety of subjective and possibly contradictory answers (see Figure 5.6, from *Amazon*).

5.3.1 Problem Definition

Similar to the problem definition in [113], the product-related questions can be categorized into two types as follows.

- **Binary questions.** A large fraction of questions in real-world opinion QA data are binary questions where answers amount to either 'Yes' or 'No'. Such answers can easily be detected (i.e., to build a labeled dataset) in a supervised setting using a binary classifier [62]. When addressing binary questions, we are interested both in mining relevant opinions from reviews, but also providing a yes/no answer directly.
- **Open-ended questions.** In addition to binary questions, a significant number of product-related questions are open-ended, or compound questions (etc.). It is usually impractical to answer such questions directly with an automated system. Instead, we are more interested in learning a good relevance function which can help us retrieve useful information from reviews, so that the user can be aided in reaching a conclusion themselves.

In this work, we continue to study these two types of questions, but explicitly account for the fact that questions may have multiple, subjective, and possibly contradictory answers.[2] Our main goals here are to show quantitatively that by leveraging multiple answers in a supervised framework we can provide more accurate responses to both subjective and objective questions (where 'accurate' for a subjective question means that we can correctly estimate the distribution of views). Addressing this new view of question-answering is challenging, and requires new techniques to be developed in order to make use of multiple, possibly contradictory labels within a supervised framework. We identify two main perspectives from which ambiguity and subjectivity in product-related opinion QA systems can be studied:

- **Multiple Answers.** We notice that in previous studies, only one ground-truth answer is included for each question. However, in real-world opinion QA systems, multiple answers are often available. We find this to be true both for binary and open-ended questions. When

[2] Note that even binary questions may still be subjective, such that both 'yes' and 'no' answers may be possible.

multiple answers are available, they often describe different aspects of the questions or different personal experiences. By including multiple answers at training time, we expect that the relevant reviews retrieved by the system at test time should cover those subjective responses more comprehensively.

- **Subjective Reviews.** In addition, as indicated in traditional opinion mining studies, reviews as reflections of users' opinions may be subjective since different reviewers may have different expertise and bias. In some review websites, such as *Amazon.com*, review rating scores and review helpfulness can be obtained, which could be good features reflecting the subjectivity of the reviews. Intuitively, subjective information may affect the language that users apply to express their opinion so that their reviews should be handled to address questions accordingly. For example, 'picky' reviewers may tend to provide negative responses while 'generous' reviewers may usually provide more favorable information about the product. This motivates us to apply user modeling approaches and incorporate more subjective review-related features into opinion QA systems.

The above observations provide us with a strong motivation to study ambiguity and subjectivity from the perspective of multiple answers and subjective reviews in opinion QA systems. We conclude by stating the problem specifically as follows:

- **Goal:** Given a question related to a product, we would like to determine how relevant each review of that product is to the question with emphasis on modeling **ambiguity** and **subjectivity**, where 'relevance' is measured in terms of how helpful the review will be in terms of identifying the proper response (or responses) to the question.

In this work, we aim to build systems that are capable of presenting users with a more nuanced selection of supporting evidence, capturing the full spectrum of relevant opinions. We evaluate our system by collecting a new QA dataset from *Amazon.com*—consisting of 800 thousand questions and 3.1 million answers, which uses *all* of the available answers for training (in contrast to previous approaches, where each question was associated with only a single answer).

5.3.2 Related Work

There are several previous studies considering the problem of opinion question answering [13–15, 98, 113, 118, 148, 171], where questions are subjective and traditional QA approaches may not be as effective as they have been for factual questions. Yu and Hatzivassiloglou [170] first proposed a series approaches to separate opinions and facts and identify the polarities of opinion sentences. Ku *et al.* [93] applied a two-layer framework to classify questions and estimated question types and polarities to filter irrelevant sentences. Li *et al.* [98] proposed a graph-based approach that regarded sentences as nodes and weighted edges by sentences similarity; by constructing such a graph, they could apply an 'Opinion PageRank' model and an 'Opinion HITS' model to explore different relations. Particularly for product-related opinion QA, i.e., addressing product-related questions with reviews, an aspect-based approach was proposed where aspect-rating data were applied [118].

In Yu *et al.* [171], a new model was developed to generate appropriate answers for opinion questions by exploiting the hierarchical organization of consumer reviews. Most recently, a supervised learning approach, *MoQA*, was proposed for the product-related opinion QA problem, where a mixture of experts model was applied and each review was regarded as an expert [113].

Opinion mining is a broad topic where customer reviews are a powerful resource to explore. A number of opinion mining studies focus on opinion summarization [71], and opinion retrieval and search in review text [102]. In addition, review text can be used to improve recommender systems by modeling different aspects related to customers' opinions [111, 156]. Subjective features and user modeling approaches were frequently applied in these studies, though they were not considered for the opinion QA problem.

The major technique of modeling ambiguity with multiple labels in this study is inspired by approaches for resolving noisy labels in crowdsourcing tasks [137]. Notice that the main target of crowdsourcing is to resolve conflicts from annotators and obtain the actual label instead of directly providing accurate predictions from data, which is different from the setting of answering

subjective questions as in our opinion QA problem. In essence, our study can be regarded as a combination of question answering, opinion mining and the idea of learning from crowds.

5.3.3 Methodology

In this study, we build upon the mixture of experts (MoE) framework as used previously by [113]. We enhance this approach by modeling ambiguity and subjectivity from the perspectives of answers and reviews. Before introducing the complete model, we introduce standard relevance measures and the mixture of experts (MoE) framework as background knowledge.

We first describe two kinds of similarity measures for relevance ranking in the context of our opinion QA problem as follows.

- **Okapi BM25.** One of the standard relevance ranking measures for information retrieval, Okapi BM25 is a bag-of-words 'tf-idf'-based ranking function that has been successfully applied in a number of problems including QA tasks [81, 109]. Particularly, for a given question q and a review r, the standard BM25 measure is defined as

$$bm25(q,r) = \sum_{i=1}^{n} \frac{idf(q_i) \times f(q_i,r) \times (k_1+1)}{f(q_i,r) + k_1 \times (1-b+b \times \frac{|r|}{avgrl})},$$

  where $q_i, i = 1,\ldots,n$ are keywords in q, $f(q_i,r)$ denotes the frequency of q_i in r, $|r|$ is the length of review r and $avgrl$ is the average review length among all reviews.[3] Here $idf(q_i)$, the inverse document frequency of q_i, is defined as

$$idf(q_i) = \log \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5},$$

  where $N = |R|$ is the total number of reviews and $n(q_i)$ is the number of reviews which contain q_i.

- **Rouge-L.** Next we consider another similarity measure, Rouge-L [99], which is a Longest Common Subsequence (LCS) based statistic. For a question q and a review r, if the

[3] In practice we set $k_1 = 1.5$ and $b = 0.75$.

length of their longest common subsequence is denoted as $LCS(q,r)$, then we have $R_{LCS} = LCS(q,r)/|q|$ and $P_{LCS} = LCS(q,r)/|r|$. Now Rouge-L is defined as

$$F_{LCS} = \frac{(1+\beta^2)R_{LCS}P_{LCS}}{R_{LCS}+\beta^2 P_{LCS}},$$

where $\beta = P_{LCS}/R_{LCS}$.

Backbone Framework: Mixtures of Experts

Mixtures of experts (**MoE**) [82] is a supervised learning approach that smoothly combines the outputs of several 'weak' classifiers in order to generate predictions. Here, this method can be applied for opinion QA systems where each individual review is regarded as a weak classifier that makes a prediction about the response to a query. For each classifier (review), we output a relevance/confidence score (how relevant is this review to the query?), as well as a prediction (e.g. is the response 'yes' based on the evidence in this review?). Then an overall prediction can be obtained for a particular question by combining outputs from all reviews of a product, weighted by their confidence.

MoE for binary questions. For a binary question, each classifier produces a probability associated with a positive label, i.e., a probability that the answer is 'yes.' Suppose for a question q, the associated features (including the text itself, the identity of the querier, etc.) are denoted X_q and the label for this question is denoted as y_q ($y_q \in \{0,1\}$). Then we have

$$P(y_q|X_q) = \sum_{r\in R_q} \overbrace{P(r|X_q)}^{\text{how relevant is } r} \times \overbrace{P(y_q|r,X_q)}^{\text{prediction from } r}, \tag{5.7}$$

where r is a review among the set of reviews R_q associated with the question q. In Eq. (5.7), $P(r|X_q)$ measures the confidence of review r's ability in terms of responding to the question q, and $P(y_q|r,X_q)$ is the prediction for q given by review r. These two terms can be modeled as

follows:

$$\begin{aligned} &\text{(Relevance)} \quad P(r|X_q) = \exp(v_{q,r}) / \sum_{r' \in R_q} \exp(v_{q,r'}); \\ &\text{(Prediction)} \quad P(y_q = 1|r, X_q) = \sigma(w_{q,r}), \end{aligned} \tag{5.8}$$

where $\sigma(x) = 1/(1+\exp(-x))$ is the sigmoid function. Here $v_{q,r}$ and $w_{q,r}$ are real-valued (i.e., unnormalized) 'relevance' and 'prediction' scores where multiple question and review related features can be involved.

MoE for open-ended questions. Similarly, for an open-ended question, we may be interested in whether a 'true' answer a_q is preferred over some arbitrary non-answer $\bar{a}$. For this we have a similar MoE structure as follows:

$$P(a_q > \bar{a}|X_q) = \sum_{r \in R_q} P(r|X_q) P(a_q > \bar{a}|r, X_q). \tag{5.9}$$

The relevance term can be kept the same while we have a slightly different prediction term:

$$P(a_q > \bar{a}|r, X_q) = \sigma(w_{a_q > \bar{a}, r}). \tag{5.10}$$

Here $w_{a_q > \bar{a}, r}$ is a real-valued 'prediction' score where multiple answer and review features can be included.

Relevance and Prediction with Text-Only Features

As described above, for a binary question, the probability associated with a positive (i.e., 'yes') label $P(y_q = 1|X_q)$ (p_q in shorthand) can be modeled using an MoE framework where each review is regarded as a weak classifier. If only one label is included for a question in the training procedure, we can train by maximizing the following log-likelihood:

$$\mathscr{L} = \log P(\mathscr{Y}|\mathscr{X}) = \sum_q \Big(y_q \log p_q + (1 - y_q) \log(1 - p_q) \Big) \tag{5.11}$$

where Θ includes all parameters and p_q is modeled as in Eq. (5.7).

A number of features can be applied to define the 'relevance' ($v_{q,r}$) and 'prediction' ($w_{q,r}$) functions. Previously in [113], only text features were used to define pairwise similarity measures and bilinear models. Starting with the same text-only model, suppose $\boldsymbol{f}_q$ and $\boldsymbol{f}_r$ are vectors with length N that represent bag-of-words text features for question q and review r. Then we define the 'relevance' function as follows:

$$v_{q,r} = \overbrace{\langle \boldsymbol{\kappa}, \boldsymbol{s}(q,r) \rangle}^{\text{pairwise similarities (bm25 etc.)}} + \overbrace{\langle \boldsymbol{\eta}, \boldsymbol{f}_q \circ \boldsymbol{f}_r \rangle}^{\text{term-to-term similarity}}, \tag{5.12}$$

where $\boldsymbol{x} \circ \boldsymbol{y}$ is the Hadamard product. Note that we have two parts in $v_{q,r}$: (1) a weighted combination of state-of-the-art pairwise similarities; and (2) a parameterized term-to-term similarity. Following [113], we include BM25 [109] and Rouge-L [99] measures in $\boldsymbol{s}(q,r)$. Recall that the purpose of this function is to learn a set of parameters $\{\boldsymbol{\kappa}, \boldsymbol{\eta}\}$ that ranks reviews in order of relevance. In addition, we define the following prediction function:

$$w_{q,r} = \overbrace{\langle \boldsymbol{\mu}, \boldsymbol{f}_q \circ \boldsymbol{f}_r \rangle}^{\text{interaction between question \& review text}} + \overbrace{\langle \boldsymbol{\xi}, \boldsymbol{f}_r \rangle}^{\text{prediction from review text}}. \tag{5.13}$$

The idea here is that the first term models the interaction between the question and review text, while the second models only the review (which can capture e.g. sentiment words in the review).

Finally, to optimize the parameters $\Theta = \{\boldsymbol{\kappa}, \boldsymbol{\eta}, \boldsymbol{\mu}, \boldsymbol{u}\}$ from Eq. (5.12) and Eq. (5.13), we apply L-BFGS [122]. To avoid overfitting, this model also includes a simple ℓ^2 regularizer on all model parameters.

Modeling Ambiguity: Learning with Multiple Labels

So far, we have introduced the basic model with text-only features, and a single label (answer) associated with each question. But as we find in our data (see Section 5.3.4), responses

in real-world opinion QA systems have significant ambiguity, even for binary questions. In this section we develop additional machinery allowing us to model ambiguity and subjectivity, and in particular to handle training data with multiple (and possibly contradictory) answers.

Notice that in the previous log-likelihood exppression Eq. (5.11), only one label can be included for each question. Below are two options to extend this framework to handle multiple labels.

KL-MoE. A straightforward approach is to replace the single label y_q in Eq. (5.11) by the the fraction of positive labels $r_q = n_q^+/(n_q^+ + n_q^-)$, where n_q^+, n_q^- are the number of positive and negative (yes/no) answers for question q. If we assume that for a question q, the response provided from the answers given follows $Bernoulli(r_q)$ and the response predicted from reviews follows $Bernoulli(p_q)$, then the objective function

$$\sum_q \Big(r_q \log p_q + (1 - r_q) \log(1 - p_q) \Big) \tag{5.14}$$

can be regarded as the summation of the KL-divergences between answers and predictions for all questions.

EM-MoE. Note that only the *ratio* of positive and negative labels is included in the previous KL-divergence loss Eq. (5.14), while the real counts of positive and negative labels are discarded. However, this fraction may not be enough to model the strength of the ambiguity (or controversy) in the question. For example, a question with 10 positive and 10 negative labels seems more controversial than a question with 1 positive and 1 negative label. However, their positive/negative ratios r_q are the same.

To distinguish such cases, instead of applying a fixed ratio r_q, we use two sets of parameters, allowing us to incorporate multiple noisy labels at training time, and to update (our noisy estimate of) r_q based on multiple labels $y_{q,j}$ and generated predictions p_q iteratively using the EM-algorithm.

Specifically, for a binary question q, we model its 'true' answer y_q as an unknown with probability distribution $P(y_q = 1|X_q, \Theta)$, which is assumed to generate the provided (noisy) labels

$y_{q,j}$ ($j = 1, \ldots, n_q$) independently. Then the joint probability of the observed labels is given by

$$\begin{aligned} P(y_{q,1}, \ldots, y_{q,n_q} | X_q, \Theta) &= \sum_{i \in \{0,1\}} P(y_{q,1}, \ldots, y_{q,n_q} | y_q = i, X_q, \Theta) P(y_q = i | X_q, \Theta) \\ &= \sum_{i \in \{0,1\}} \Big(\prod_{j=1}^{n_q} P(y_{q,j} | y_q = i, X_q, \Theta) \Big) P(y_q = i | X_q, \Theta) \end{aligned} \tag{5.15}$$

Here we separate the joint probability into two parts: (1) $P(y_q = i | X_q, \Theta)$ models the estimated distribution of the 'true' answer y_q from the provided reviews. (2) $\prod_{j=1}^{n_q} P(y_{q,j} | y_q = i, X_q, \Theta)$ models the probability of a given ground-truth label $y_{q,j}$ as a function of y_q.

Letting $\alpha_q = P(y_{q,j} = 1 | y_q = 1, X_q, \Theta)$ and $\beta_q = P(y_{q,j} = 0 | y_q = 0, X_q, \Theta)$ for all $j \in \mathscr{S}_q$, then α_q and β_q represent the 'sensitivity' (probability of a positive observation if the true label is positive) and 'specificity' (probability of a negative observation if the label is negative) for question q. Note that 'positive' and 'negative' questions may not be symmetric concepts (i.e., different types of questions may be more likely to have yes vs. no answers). Thus we model sensitivity and specificity separately, using features from the question text as prior knowledge. Specifically, we model α and β as:

$$\alpha_q = \sigma(\langle \boldsymbol{\gamma_1}, \boldsymbol{f}_q \rangle); \qquad \beta_q = \sigma(\langle \boldsymbol{\gamma}_2, \boldsymbol{f}_q \rangle). \tag{5.16}$$

Then we have the following joint distributions which are denoted as a_q and b_q:

$$\begin{aligned} a_q &:= \prod_{j=1}^{n_q} P(y_{q,j} | y_q = 1, X_q, \Theta) = \alpha_q^{n_q^+} (1 - \alpha_q)^{n_q^-} \\ b_q &:= \prod_{j=1}^{n_q} P(y_{q,j} | y_q = 0, X_q, \Theta) = (1 - \beta_q)^{n_q^+} \beta_q^{n_q^-}. \end{aligned} \tag{5.17}$$

Now based on Eq. (5.15), Eq. (5.16), and Eq. (5.17), we can consider maximizing following

log-likelihood:

$$\begin{aligned}\mathscr{L} &= \log P(\mathscr{Y}|\mathscr{X},\Theta) = \sum_q \log P(y_{q,j}, j=1,\ldots,n_q|X_q) \\ &= \sum_q \log\Big(a_q p_q + b_q(1-p_q)\Big),\end{aligned} \tag{5.18}$$

where $p_q = P(y_q = 1|X_q,\Theta)$ is modeled based on Eq. (5.7), Eq. (5.8), Eq. (5.12) and Eq. (5.13). Here the parameter set is $\Theta = \{\boldsymbol{\kappa}, \boldsymbol{\eta}, \boldsymbol{\mu}, \boldsymbol{u}, \boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2\}$.

In contrast to **MoE** and **KL-MoE**, directly optimizing Eq. (5.18) is non-trivial. However, we can apply the EM Algorithm [41] to optimize it by estimating the label y_q and the parameters Θ iteratively. By introducing the missing labels $\{y_q\}$, we have a complete likelihood expression

$$\mathscr{L}_c = \sum_q \Big(y_q \log a_q p_q + (1-y_q)\log b_q(1-p_q)\Big). \tag{5.19}$$

- In the **E-step**, we assume that parameters Θ are given. Then we take the expectation of y_q in Eq. (5.19) and we obtain a new objective:

  $$\mathbb{E}\mathscr{L}_c = \sum_q \Big(t_q \log a_q p_q + (1-t_q)\log b_q(1-p_q)\Big),$$

  where

  $$t_q = P(y_q = 1|y_{q,1},...,y_{q,n_q},X_q) = \frac{a_q p_q}{a_q p_q + b_q(1-p_q)}.$$

- In the **M-step**, once t_q is obtained, similar to **MoE** and **KL-MoE**, we can apply L-BFGS [122] to optimize $\mathbb{E}\mathscr{L}_c$ with respect to Θ.

These two procedures are repeated until convergence.

Incorporating Subjective Information

EM-MoE-S. Subjective information from reviews (and review*ers*) can be included to enhance the performance of both our relevance and prediction functions, including features such as review helpfulness, reviewer expertise, rating scores and reviewer biases. We can incorporate

these features into our previous expressions for $v_{q,r}$ and $w_{q,r}$ as follows:

$$v_{q,r} = \overbrace{\langle \boldsymbol{\kappa}, \boldsymbol{s}(q,r) \rangle}^{\text{pairwise similarities}} + \overbrace{\langle \boldsymbol{\eta}, \boldsymbol{f}_q \circ \boldsymbol{f}_r \rangle}^{\text{term-to-term similarity}} + \overbrace{\langle \boldsymbol{g}, \boldsymbol{h}_r \rangle}^{\text{review's helpfulness}} + \overbrace{e_{u_r}}^{\text{reviewer's expertise}}$$

$$w_{q,r} = (\underbrace{\langle \boldsymbol{\mu}, \boldsymbol{f}_q \circ \boldsymbol{f}_r \rangle}_{\text{interaction bet. q. \&r. text}} + \underbrace{\langle \boldsymbol{\xi}, \boldsymbol{f}_r \rangle}_{\text{prediction from r. text}}) \times (1 + \underbrace{c \cdot rt_r}_{\text{rating score}} + \underbrace{b_{u_r}}_{\text{reviewer's bias}}). \tag{5.20}$$

As shown in Figure 5.6, here rt_r is the star rating score and $\boldsymbol{h}_r = (h_r^{(1)}, h_r^{(2)})^T$ represents the helpfulness features of review r where $h_r^{(1)}, h_r^{(2)}$ are fractions of users who respectively find or do not find the review helpful. e_{u_r} and b_{u_r} are parameters that make up a simple user model; the former captures the overall tendency for a user u to write reviews that are likely to be 'relevant,' while the latter captures the tendency of their reviews to support positive responses. Note that both parameters are latent variables that are automatically estimated when we optimize the likelihood expression above.

Modeling Open-Ended Questions

Although our **KL-MoE** and **EM-MoE** frameworks can model ambiguity in binary questions, and account for simple features encoding subjectivity, we still need to develop methods to account for ambiguity in open-ended questions. Here we are no longer concerned with divergence between yes/no answers, but rather want to model the idea that there is a pool of answers to each question which should be regarded as more valid than alternatives. As with binary questions, these open-ended questions may be subjective and multiple answers often exist in our data. What is different is that it is difficult for us to automatically judge whether these answers are consistent or not. Thus we aim to generate candidate answers that cover the spectrum of ground-truth answers as much as possible.

First we give some more detail about the basic framework with a single open-ended

answer, which we described briefly in Section 5.3.3. Then we simply extend this framework to include multiple open-ended answers and incorporate subjective information.

Single Answer (s-MoE). Our objective for open-ended questions is to maximize the Area Under Curve (AUC), which is defined as

$$AUC_o = \frac{1}{|Q|}\sum_q AUC(q) = \frac{1}{|Q|}\sum_q (\frac{1}{|\bar{\mathscr{A}_q}|}\sum_{\bar{a}\in\bar{\mathscr{A}_q}} \delta(a_q > \bar{a})).$$

where a_q is the ground-truth answer to the question q and $\bar{\mathscr{A}_q}$ is a set of non-answers (randomly sampled from among all answers). In other words, a good system is one that can correctly determine which answer is the real one.[4] In practice, we maximize a smooth objective to approximate this measure in the form of the log-likelihood:

$$\mathscr{L} = \sum_q \sum_{\bar{a}\in\bar{\mathscr{A}_q}} \log p_{q,a_q>\bar{a}}.$$

Here $p_{q,a_q>\bar{a}} = P(a_q > \bar{a}|X_q)$ is as defined in Eq. (5.9). The 'relevance' term in $p_{q,a_q>\bar{a}}$ is the same as for binary questions while the 'prediction' term is defined as

$$p_{q,a_q>\bar{a}|r} = \sigma(w_{a_q>\bar{a}|r}).$$

As before, $w_{a_q>\bar{a}|r}$ can be modeled in terms of answer and review text. Letting $\boldsymbol{f}_{a_q}$ and $\boldsymbol{f}_{\bar{a}}$ denote the text features of the answer a_q and the non-answer $\bar{a}$ respectively. Then we have

$$w_{a_q>\bar{a}} = w_{a_q,r} - w_{\bar{a},r} = \overbrace{\left\langle \boldsymbol{\mu}, (\boldsymbol{f}_{a_q} - \boldsymbol{f}_{\bar{a}}) \circ \boldsymbol{f}_r \right\rangle}^{\text{interaction between ans. difference \& review text}}. \tag{5.21}$$

$\boldsymbol{f}_{a_q} - \boldsymbol{f}_{\bar{a}}$ represents the difference between the answers a_q and $\bar{a}$, so that Eq. (5.21) models which of the answers a_q or $\bar{a}$ is *more supported* by review r.

Multiple Answers (m-MoE). The previous AUC measure can be straightforwardly extended to be compatible with multiple answers. If multiple answers exist for a question q, then

[4]Note that in practice, at test time, one would not have a selection of candidate answers to choose from; the purpose of the model in this case is simply to identify which reviews are relevant (by using the answers at *training* time), rather than to answer the question directly.

our target is to maximize the following AUC measure:

$$AUC_o = \frac{1}{|Q|}\sum_q (\frac{1}{|\mathscr{A}_q||\bar{\mathscr{A}}_q|}\sum_{a\in\mathscr{A}_q}\sum_{\bar{a}\in\bar{\mathscr{A}}_q}\delta(a>\bar{a})). \tag{5.22}$$

where $\mathscr{A}_q$ denotes the *set* of answers to question q and $\bar{\mathscr{A}}_q$ is defined as before. Similarly, we maximize the following log-likelihood loss function to approximately optimize the AUC:

$$\mathscr{L} = \sum_q \frac{1}{|\mathscr{A}_q|}\sum_{a\in\mathscr{A}_q}\sum_{\bar{a}\in\bar{\mathscr{A}}_q}\log p_{q,a>\bar{a}}. \tag{5.23}$$

Incorporating Additional Information from Reviews (m-MoE-S). Similar to binary questions, we can incorporate more subjective features into $v_{q,r}$ and $w_{a>\bar{a},r}$. Basically, $v_{q,r}$ can be kept the same as in Eq. (5.20). For $w_{a_q>\bar{a},r}$, we have

$$w_{a_q>\bar{a},r} = \overbrace{\underbrace{\left\langle \boldsymbol{\mu}, (\boldsymbol{f}_{a_q} - \boldsymbol{f}_{\bar{a}}) \circ \boldsymbol{f}_r \right\rangle}_{\text{interaction b.w. ans. difference \&r. text}}}^{\text{which answer the review favors}} \times \overbrace{(1 + \underbrace{c \cdot rt_r}_{\text{rating score}} + \underbrace{b_{u_r}}_{\text{reviewer's bias}})}^{\text{how supportive based on the review}}. \tag{5.24}$$

The left part of this formula is the same as in Eq. (5.21) which models which of the answers the review favors. The right part of this formula is an amplifier which models how supportive the review r is based on its subjective information.

5.3.4 Dataset and Exploratory Analysis

In [113], the authors collected Q/A data from *Amazon.com*, including a single answer (the top-voted) for each question. We collected all the related urls in this dataset and further crawled all available answers to each question (duplicates were discarded, as were questions that have been removed from *Amazon* since the original dataset was collected). For each product we also have its related reviews. Ultimately we obtained around 808 thousand questions with 3 million answers on 135 thousand products in 8 large categories. For these products, we have 11 million reviews in total. Detailed information is shown in Table 5.3.

Table 5.3: Basic statistics of our Amazon dataset.

Category	#products	#questions	#answers	#reviews
Automotive	10,578	59,449	233,784	325,523
Patio, Lawn & Garden	7,909	47,589	193,780	450,880
Tools & Home Improv.	13,315	81,634	327,597	751,251
Sports & Outdoors	19,102	114,523	444,900	988,831
Health & Personal Care	10,766	63,985	255,209	1,154,315
Cell Phones	10,320	60,791	237,220	1,353,441
Home & Kitchen	24,329	148,773	611,335	2,007,847
Electronics	38,959	231,556	867,921	4,134,100
Total	135,278	808,300	3,171,746	11,166,188

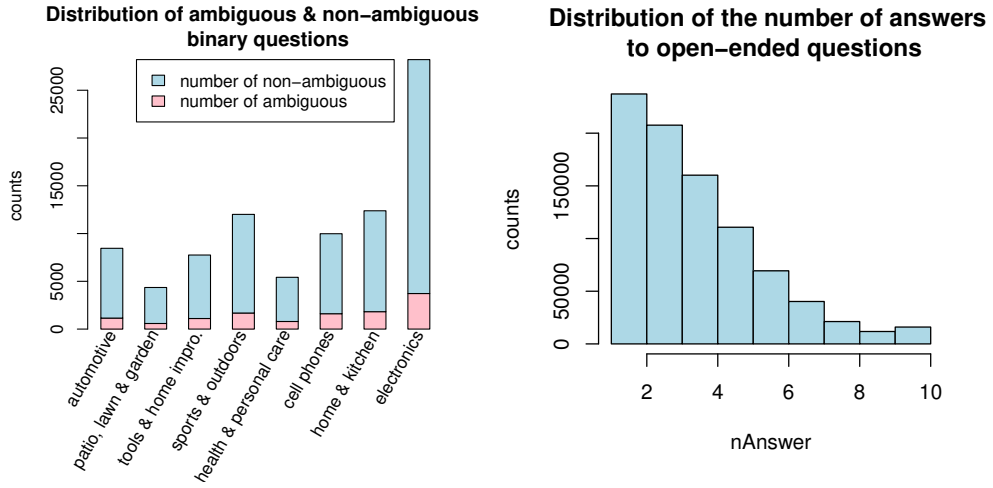
In practice, we split review paragraphs into sentences, such that each sentence is treated as a single 'expert' in our MoE framework. We used the Stanford CoreNLP [110] library to split reviews into sentences, handle word tokenization, etc.

Ground-Truth Labels for Binary Questions

In the dataset from [113], one thousand questions have been manually labeled as 'binary' or 'open-ended.' For binary questions, a positive or negative label is provided for each answer. We used these labels as seeds to train simple classifiers to identify binary questions with positive and negative answers. We applied an approach developed by *Google* [63] to determine whether a question is binary, using a series of simple grammatical rules. Among our labeled data, this approach achieved 97% precision and 82% recall in this manually labeled dataset.[5]

Following this, we developed a simple logistic regression model to label observed answers to these binary questions. The features we applied are the frequency of each unigram plus whether the first word is 'yes' or whether it is 'no' (as is often the case in practice). Notice that since we want to study ambiguity that arises due to the question itself rather than due to any error in our machine labels, we need to ensure that the binary labels obtained from this logistic model are

[5]Note that we are happy to sacrifice some recall for the sake of precision, as low recall simply means discarding some instances from our dataset, as opposed to training on incorrectly labeled instances.



(a) Distribution of ambiguous and non-ambiguous binary questions.

(b) Histogram of numbers of answers for all open-ended questions

Figure 5.7: Distribution of the dataset.

as accurate as possible. Thus again we sacrifice some recall and keep only those answers about which the regressor is most confident (here we kept the top 50% of most confident predictions). This gave us zero error on the held-out manually labeled data from [113]. Ultimately we obtained 88,559 questions with 197,210 highly confident binary labels of which around 65% are positive. In our experiments, two thirds of these questions and associated labels are involved in training and the rest are used for evaluation.

Exploratory Analysis

Having constructed classifiers to label our training data with high confidence, we next want to determine whether there really are conflicts between multiple answers for binary questions. The distribution of ambiguous (i.e., both yes and no answers) versus non-ambiguous binary questions is shown in Figure 5.7a. From this figure, we notice that we do have a portion of binary questions that can be confidently classified as 'ambiguous.' A real-world example of such a question is shown in Figure 5.6. One might expect that the answer to such a question would be an unambiguous 'yes' or 'no,' since it is a (seemingly objective) question about compatibility.

However the answers prove to be inconsistent since different users focus on different aspects of the product. Thus even seemingly objective questions can prove to be 'ambiguous,' demonstrating the need for a model that handles such conflicting evidence. Ideally a system to address such a query would retrieve relevant reviews covering a variety of angles and in this case provide an overall neutral prediction.

Ultimately, around 14% of the questions in our dataset are ambiguous (i.e., multiple binary labels are inconsistent). Distributions of ambiguous/non-ambiguous questions are plotted in Figure 5.7a. Even though we filtered our dataset to include only answers with high-confidence labels (i.e., clear 'yes' vs. 'no' answers), there is still a significant number of questions with conflicting labels, which indicates that modeling ambiguity is necessary for opinion QA systems.

5.3.5 Experiments

We evaluate our proposed methods for binary questions and open-ended questions on a large dataset composed of questions, answers and reviews from *Amazon*. For binary questions, we evaluate the model's ability to make proper predictions. For open-ended questions, we evaluate the model's ability to distinguish 'true' answers from alternatives. Since our main goal is to address ambiguity and subjectivity, we focus on evaluating our model's ability to exploit multiple labels/answers, and the effect of features derived from subjective information.

Binary Questions

For yes/no questions, our target is to evaluate whether our model can predict their 'true' labels correctly. Since multiple labels are collected for a single question, and since we are comparing against methods capable of predicting only a single label, it is difficult to evaluate which system's predictions are most 'correct' in the event of a conflict. Thus for evaluation we build two test sets consisting of decreasingly ambiguous questions. Our hope then is that by modeling ambiguity and personalization during training, our system will be more reliable even

for unambiguous questions. We build two evaluation sets as follows:

- **Silver Standard Ground-truth.** Here we simply regard the majority vote among ambiguous answers as the 'true' label (questions with ties are discarded).
- **Gold Standard Ground-truth.** More aggressively, here we ignore all questions with conflicting labels. For the remaining questions, we have consistent labels that we regard as ground-truth.

Notice that all the questions and labels (ambiguous or otherwise) in the training set are involved in the training procedure of **KL-MoE**, **EM-MoE** and **EM-MoE-S**. We only attempt to resolve ambiguity when building our test set for evaluation.

Naturally, it is not possible to address all questions using the content of product reviews. Thus we are more interested in the probability that the model will rank a random positive instance higher than a random negative one. We adopt the standard classification AUC measure to evaluate the model performance where a naïve classifier (random predictions, random confidence ranks) has an AUC of 0.5. We compare the performance of the following methods:

- **MoE.** This is a state-of-the-art method for opinion QA from [113]. This is the model described in Section 5.3.3. Here only a single label (the top-voted) is used for training, and text features from the reviews are included.
- **KL-MoE.** This is a straightforward approach to include multiple labels by replacing a single label y_q by the ratio of positive vs. negative answers ($r_q = n_q^+ / (n_q^+ + n_q^-)$ in Eq. (5.11) (see Sec. 5.3.3).
- **EM-MoE.** To include all the labels instead of just a ratio, we use an EM-like approach to update our estimates of noisy labels and parameters iteratively. Here question text features are used as prior knowledge to model the 'sensitivity' and 'specificity' regarding a question.
- **EM-MoE-S.** Note that the above models only make use of features from reviews, and are designed to measure the performance improvements that can be achieved by harnessing multiple labels. For our final method, we include other subjective information into our

model, such as user bieses, rating features, etc. (see Sec. 5.3.3).

Ultimately the above baselines are intended to demonstrate: (a) the performance of the existing state-of-the-art (**MoE**); (b) the improvement from leveraging conflicting labels during training (**KL-MoE** and **EM-MoE**); and (c) the improvement from incorporating additional subjective information in the data (**EM-MoE-S**).

Results of the above methods in terms of the AUC are shown in Table 5.4. We notice that while **KL-MoE** is not able to improve upon **MoE** for all categories, **EM-MoE** and **EM-MoE-S** yield consistent improvements in all cases.[6] This improvement is relatively large for some large categories, such as *Cell Phones & Accessories* and *Electronics*. Incorporating subjective features (**EM-MoE-S**) seems to help most for large categories, indicating that it is useful when enough training data is available to make the additional parameters affordable.

When modeling ambiguity in opinion QA systems, a possible reason for the failure of **KL-MoE** is that the ratio r_q involved in the objective function may not be a representative label for training. If the observed positive label ratio r_q does not properly reflect the 'true' distribution, it could adversely affect the optimization procedure. In our EM-like frameworks, i.e., **EM-MoE** and **EM-MoE-S**, this ratio is replaced by a posterior probability, t_q, which is updated iteratively. These EM-like frameworks are relatively more robust to data with multiple noisy labels compared with **KL-MoE**. **EM-MoE-S** includes subjective information related to reviews and reviewers. Due to the number of parameters involved, modeling reviewer expertise and bias is only useful for users who write several reviews, which is indeed a small fraction of reviewers. Thus in the larger categories these terms appear more useful, once we have enough observations to successfully model them.

Note that the AUC represents the ranking performance on all questions. Generally, this value is relatively low in our experiments. This is presumably due to the simple fact that many questions cannot be answered based on the evidence in reviews. Since all of the methods being

[6] Improvements in accuracy over **MoE** are statistically significant at the 1% level or better.

Table 5.4: Results on binary questions where multiple noisy labels are involved.

(a) Silver Standard Ground-truth

	MoE	**KL-MoE**	**EM-MoE**	**EM-MoE-S**
Automotive	0.5226	0.5326	**0.5354**	0.5225
Patio Lawn & Garden	0.5010	0.5184	**0.5257**	0.5173
Tools & Home Improv.	0.5514	0.5313	**0.5690**	0.5641
Sports & Outdoors	0.5536	0.5512	0.5567	**0.5578**
Health & Personal Care	0.5405	0.5157	0.5490	**0.5588**
Cell Phones	0.5612	0.5506	0.5936	**0.6012**
Home & Kitchen	0.5087	0.5027	0.5130	**0.5394**
Electronics	0.5525	0.5172	0.5966	**0.6002**

(b) Gold Standard Ground-truth

	MoE	**KL-MoE**	**EM-MoE**	**EM-MoE-S**
Automotive	0.5218	0.5363	**0.5415**	0.5285
Patio Lawn & Garden	0.5030	0.5238	**0.5271**	0.5124
Tools & Home Improv.	0.5511	0.5280	**0.5627**	0.5547
Sports & Outdoors	0.5538	0.5491	0.5587	**0.5628**
Health & Personal Care	0.5452	0.5166	0.5530	**0.5621**
Cell Phones	0.5661	0.5534	0.5984	**0.6062**
Home & Kitchen	0.5115	0.5052	0.5165	**0.5382**
Electronics	0.5540	0.5171	0.5983	**0.6046**

compared output confidence scores, we are interested in whether competing systems are correct in those instances where they have high confidence. If Q denotes the set of all the questions and Q_a denotes the set of questions associated with the first largest $(1-a)|Q|$ values of $|\hat{p}_q - 0.5|$ (i.e., the most confident about *either* a yes or a no answer), then we have the following measure for a given confidence threshold $0 \leq a \leq 1$:

$$accuracy@a = \frac{1}{|Q_a|} \sum_{q \in Q_a} (\mathbf{1}_{\hat{p}_q \geq 0.5, y_q = 1} + \mathbf{1}_{\hat{p}_q < 0.5, y_q = 0}).$$

Recall that the AUC measures the model's ability to rank questions appropriately based on the ground-truth positive and negative labels. In contrast, the *accuracy@a* instead measures the model's ability to correctly predict labels of those questions with highly confident output ranks. We plot this accuracy score as a function of a for the smallest category (*Automotive*) and the

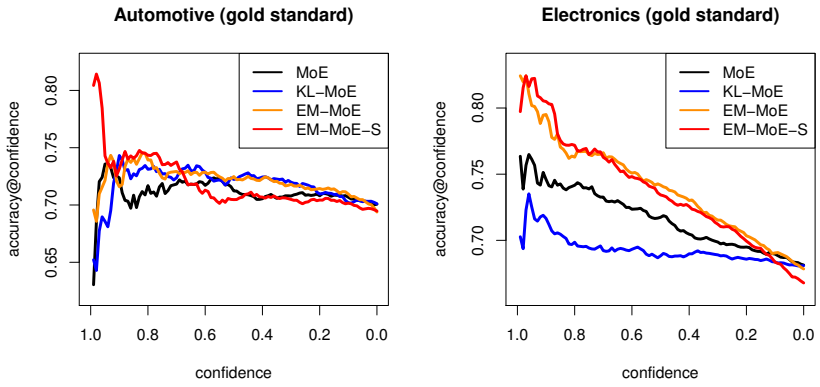


Figure 5.8: Accuracy as a function of confidence on binary questions (Automotive and Electronics categories).

largest category (*Electronics*) in Figure 5.8. We notice that the improvement from modeling ambiguity (**MoE** vs. others) is relatively consistent for all confidence levels. However, modeling subjective information only seems to improve the performance on the most highly confident instances. For a small category like *Automotive*, since there is too little data to model those inactive reviewers, the **EM-MoE-S** model performs poorly on low-confidence instances.

Open-ended Questions

After our previous procedure to distinguish binary vs. open-ended questions, we are left with a total of 698,618 open-ended questions (85% of all the questions) in our dataset. We plot the distribution of the number of answers provided to each open-ended question in Figure 5.7b and find that the majority of these questions have more than one answer provided. Our goal here is to explore whether using multiple answers at training time can provide us with more accurate results, in terms of the AUC of Eq. (5.22). In practice, for each answer a, we randomly sample one alternative non-answer $\bar{a}$ from the pool of all answers. Suppose the output probability that answer a to question q is preferred over a non-answer $\bar{a}$ is $\hat{p}_{q,a>\bar{a}}$. Then the AUC measure is

Table 5.5: Results on open-ended questions in terms of AUC where multiple answers are involved.

	s-MoE	**m-MoE**	**m-MoE-S**
Automotive	**0.8470**	0.8446	0.8459
Patio Lawn & Garden	0.8640	**0.8737**	0.8673
Tools & Home Improv.	0.8676	**0.8760**	0.8680
Sports & Outdoors	0.8624	**0.8671**	0.8654
Health & Personal Care	0.8697	**0.8801**	0.8218
Cell Phones & Accessories	0.8326	**0.8372**	0.8232
Home & Kitchen	0.8702	**0.8746**	0.8723
Electronics	0.8481	**0.8500**	0.8480

defined as

$$AUC_o = \frac{1}{|Q|}\sum_q \frac{1}{|\mathscr{A}_q|}\sum_{a\in\mathscr{A}_q} \mathbf{1}(\hat{p}_{q,a>\bar{a}} > 0.5).$$

Note that although different answers are involved in the training procedures for different models, this evaluation measure is calculated in the same format for the same test data. We compare the performance of the following methods:

- **s-MoE.** This is the method from [113]. Here only the top-voted answer is included for training.
- **m-MoE.** We include all answers for each question in this method and optimize the objective function in Eq. (5.23). Thus we evaluate whether training with multiple answers improves performance.
- **m-MoE-S.** Similarly, we add additional subjective information to our model in order to evaluate the contribution of subjective features.

Again our evaluation is intended to compare (a) the performance of the existing state-of-the-art (**s-MoE**); (b) the improvement when training with multiple answers (**m-MoE**); and (c) the impact of including subjective features in the model (**m-MoE-S**).

Results from **s-MoE**, **m-MoE** and **m-MoE-S** are included in Table 5.5. We find that including multiple answers in our training procedure helps us to obtain slightly better results, while incorporating subjective information was not effective here. A possible reason could be that

open-ended questions may not be as polarized as binary questions so that subjective information may not be as good an indicator as compared to the content of the review itself.

5.4 Conclusions and Future Work

In Section 5.2, we presented **PRAST**, a one-class recommendation framework that allows us to make use of asymmetric textual information in implicit feedback settings. In order to overcome the challenge of asymmetry (i.e., side information that is only available for *positive* instances), we introduced a new optimization criterion incorporating a language model where preference factors and textual topics are matched in a relevance-aware way. The principle of **PRAST** can be extended to incorporate other types of 'positive-only' side information (e.g. transaction timestamps and geo-tags, review helpfulness, product prices in transaction logs, etc.). As future work, these asymmetric signals can be modeled with the proposed pairwise ranking criterion, and potentially serve as informative context for better recommendations.

In Section 5.3, we systematically developed a series of methods to model ambiguity and subjectivity in product-related opinion question answering systems. We proposed an EM-like mixture-of-experts framework for binary questions which can successfully incorporate multiple noisy labels and subjective information. Results indicate that this kind of framework consistently outperforms traditional frameworks that train using only a single label. For open-ended questions, we similarly found that including multiple answers during training improves the ability of the model to identify correct answers at test time.

5.5 Acknowledgements

This chapter is based on the materials as they appear in the *IEEE International Conference on Data Mining*, 2016 ("Modeling Ambiguity, Subjectivity, and Diverging View-points in Opinion

Question Answering Systems," Mengting Wan and Julian McAuley), and the *SIAM International Conference on Data Minin*, 2018 ("One-Class Recommendation With Asymmetric Textual Feedback," Mengting Wan and Julian McAuley). The dissertation author was the primary investigator and author of these papers.

Chapter 6

Addressing Bias and Fairness in Modeling Consumer Behavior

6.1 Introduction

By connecting users to relevant products across the vast range available on e-commerce platforms, modern recommender systems are already ubiquitous and critical on both sides of the market, i.e., consumers and product sellers. Among recommendation algorithms used in practice, many fall under the umbrella of *collaborative filtering* [68, 90, 100, 141], which collect and generalize users' preference patterns from logged interactions (e.g. purchases, ratings). These feedback interactions can be *biased* by multiple factors, potentially surfacing unfair (or irrelevant) recommendations to users or items underrepresented in the input data. Such phenomena have already raised some attention from the recommender system community: a handful of types of algorithmic biases have been addressed, including selection bias [143], popularity bias [164], and several fairness-aware recommendation algorithms have been proposed [20, 26]. In this paper, we focus on a relatively underexplored factor—*marketing bias*—in consumer-product interaction data, and study how recommendation algorithms respond to its effect.

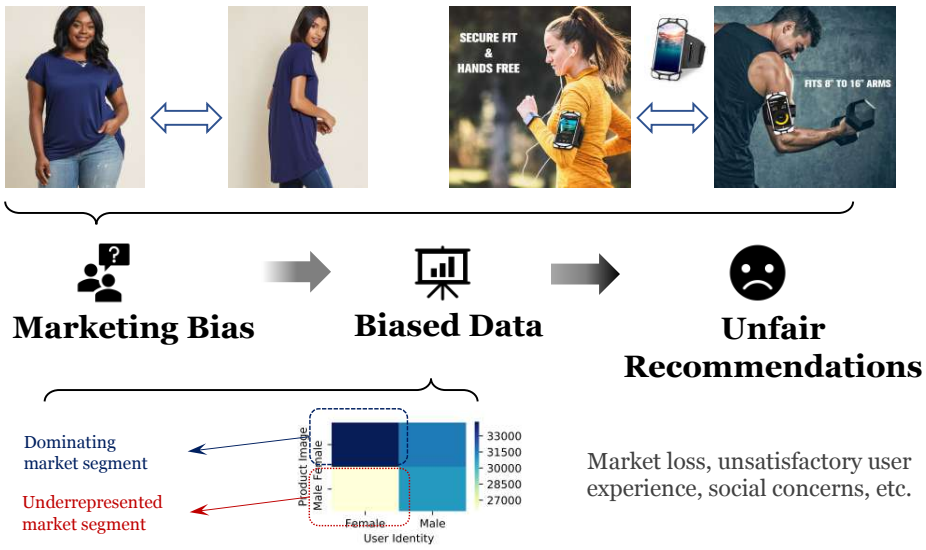


Figure 6.1: Two illustrative examples on how the same product can be marketed using different human images (body shapes/genders). These marketing strategies could affect consumers' behavior thus resulting in a biased interaction dataset, which is commonly used as the input for recommender systems.

We are particularly interested in the human factors, such as the profile of the human model in a product image, reflected in a product's marketing strategies, which (as indicated in previous marketing studies) could possibly affect consumers' interactions and satisfaction [21, 57, 58]. A common hypothesis (known as 'self-congruence') is that a consumer may tend to buy a product because its public impression (in our case a *product image*), among other alternatives, is *consistent* with one's self-perceptions (*user identity*) [58]. Based on this assumption, the selection of human models for a product (as shown in Figure 6.1, a product can be represented by models with different body shapes or different genders) could influence a consumer's behavior. For example, a female user may be less likely to interact with an armband product which is presumably gender-neutral but marketed exclusively via 'male' images. As with many other types of bias, this could lead to underrepresentation of some niche market segments in the input data for a recommender system. Note if undesired patterns are propagated into recommendation results (e.g. even fewer male-represented products are recommended to the potential female users), utility from both sides of the marketplace could be harmed. That is, product retailers may lose potential consumers

while users may be struggling to find relevant products. As a consequence, serious ethical and social concerns could be raised as well.

Summary of Contributions

In this work, we seek to understand (1) if such a marketing bias exists in real-world e-commerce datasets; (2) how common collaborative filtering algorithms interact with these potentially biased datasets; and (3) how to alleviate such algorithmic bias (if any) and improve the market fairness of recommendations. Respectively, we conclude our contributions as the follows.

We collect and process two e-commerce datasets from ModCloth[1] and Amazon[2]. Then we conduct an observational study to investigate the relationship between interaction feedback and product images (reflected in the selection of a human model) as well as user identities. Different types of correlations in varying degrees can be observed in these two datasets.

We implement several common collaborative filtering algorithms and study their responses to the above patterns in the input data. For most algorithms, we find (1) systematic deviations across different consumer-product segments in terms of rating prediction error, and (2) notable deviations of the resulting recommendation outputs from the real interaction data.

Note that as the marketing bias could be intricately entangled with users' intrinsic preferences, our goal in this work is not to pursue the absolute parity of recommendations (e.g. keep recommending products represented by human images which were constantly unfavored by a user). Rather, we expect a fair algorithm is supposed *not to worsen the market imbalance* in interactions. We thus propose a fairness-aware framework to address it by calibrating the parity of prediction *errors* across different market segments. Quantitative results indicate that our framework significantly improves recommendation fairness and provides better accuracy-fairness trade-off against several baselines.

[1] https://www.modcloth.com/
[2] https://www.amazon.com/

6.2 Related Work

This work is partially motivated by the well-known 'self-congruity' theory in marketing research, which is defined as the match between the product/brand image and the consumer's true identity and the perception about oneself [21, 57, 58]. Many previous marketing studies focus on assessing this theory by quantifying and validating it through statistical analysis on a small amount purchase transaction data or the feedback in questionnaires [92, 107, 146, 147]. Following self-congruity theory, products can be advertised in a way to match their target consumers' images thus establishing product stereotypes [55]. Our work is distinguished with these studies from a more computational perspective, by identifying and studying the potential marketing bias for recommender systems on large-scale e-commerce interaction datasets.

Our analysis is related to previous work which examines particular types of biases in real-world interactions and their effects in recommendation algorithms, including the popularity effect and catelog coverage [74], the bias regarding the book author gender for book recommenders [47], and the herding effect in product ratings [174].

Another closely related line of work includes developing evaluation metrics and algorithms to address fairness issues in recommendations. 'Unbiased' recommender systems with missing-not-at-random training data are developed by considering the propensity of each item [79, 143]. A fairness-aware tensor-based algorithm is proposed to address the absolute statistical parity (i.e., items are expected to be presented at the same rate across groups) [182]. Several fairness metrics and their corresponding algorithms are proposed for both pointwise prediction frameworks [26, 166] and pairwise ranking frameworks [20]. Methodologically, these algorithms can be summarized as reweighting schemes where underrepresented samples are upweighted [26, 79, 143] or regularizing schemes where additional fairness terms are added to regularize the model [4, 20, 166].

Note that most of the above studies focus on bias and fairness on one side of the market

Table 6.1: Basic statistics of the ***ModCloth*** and ***Electronics*** datasets.

	ModCloth	**Electronics**
#review	99,893	1,292,954
#item	1,020	9,560
#user	44,783	1,157,633
time span	2010-2019	1999-2018
bias type	body shape	gender
product image	Small (838) Small&Large (182)	Female (4,090) Female&Male (2,466) Male (3,004)
user identity	Large (9,395) Small (30,140) N/A (5,248)	Female (71,043) Male (61,350) N/A (1,025,240)

only (i.e., either user or producer). Our concern about marketing bias is that it could affect fairness for both consumers and product providers. Without global market fairness in mind, the imbalance of the consumer-product segment distribution could be exacerbated through the deployment of recommendation algorithms. Multi-sided fairness is addressed by Burke *et al* [20] by considering C(onsumer)-fairness and P(rovider)-fairness. However the CP-fairness condition where fairness is protected for both sides *at the same time* still remains an open question.

6.3 Data Collection and Preprocessing

We introduce two real-world e-commerce datasets collected from a women's clothing website *ModCloth*[3] and the *Electronics* category on *Amazon*.[4] These datasets enable us to study the marketing bias induced by the selection of a human model with respect to *body shape* for clothing products, and investigate the effects from the *gender* of human models for electronics products. These datasets will be made available at publication time. Detailed information about these datasets can be found in Table 6.1.

[3] `https://www.modcloth.com/`
[4] `https://www.amazon.com/`

Note that our datasets are not perfect, e.g. errors and selection bias can be introduced via scraping, parsing and processing, control of several confounding factors including inventory status, (etc.). Our intention here is neither to make any normative claims regarding the distributions in the above two applications, nor draw any causal conclusions. Rather, we simply describe the current state of these datasets and *study how recommendation algorithms interact with these data.*

6.3.1 ModCloth

ModCloth is an e-commerce website which sells women's clothing and accessories. One unique property of this data is that many products include two human models with different body shapes (as shown in Figure 6.1) and measurements of these models. In addition, users can optionally provide the product sizes they purchased and fit feedback ('Just Right', 'Slightly Larger', 'Larger', 'Slightly Smaller' or 'Smaller') along with their reviews. Therefore we focus on the dimension of *human body shape* as the source of marketing bias in this dataset.

Product Image Group (Body Shape)

We start with the clothing products included in an existing public dataset [117], re-scrape their landing pages, collect related model size measurements and all review ratings. We normalize their product sizes as 'XS', 'S', 'M', 'L', 'XL', '1X', '2X', '3X' and '4X' according to the provided size charts.[5] Products with only one human model wearing a relatively small size ('XS', 'S', 'M' or 'L') are labeled as the 'Small' group while products with two models (an additional model wearing a plus-size: 1X', '2X', '3X' or '4X') are referred as the 'Small&Large' group.

User Identity Group (Body Shape)

We then calculate the average size each user purchased and classify users into 'Small' and 'Large' groups based on the same standard as the product body shape image.

[5] e.g. `https://www.modcloth.com/size-guide.html`

We observe that all products offer the complete spectrum of sizes, while 70% of these products are interacted with by at least one user from the 'Large' group and 97% are interacted with by the 'Small' group. Thus we conclude that most users are able to consume most products at some point within the time frame of our dataset.

Ultimately we collect nearly 100K reviews about 1,020 clothing products from 44,783 users, where around 90% of users can be matched to the above identity groups.

6.3.2 Electronics

Electronics is another review dataset collected from the *Electronics* category on Amazon with *Clothing* as an auxiliary category. We regard the *gender* as the target marketing bias on this dataset.

Product Image Group (Gender)

We collect all pictures associated with the electronic products[6] and run human model detection through an industrial body/face detection API provided by Face++.[7] The results include whether any human bodies/faces are included in the pictures, as well as gender predictions of these detected models. We only keep products where human models are detected in their associated pictures and treat them as three types of product gender image based on the selection of these human models: 'Female' (only female models are included), 'Male' (only male models are included) and 'Female & Male' (both female and male models are detected, not necessarily in the same picture).

We then involve 3 human labelers to conduct validations on this dataset, where label conflicts are resolved by majority voting. 3,000 randomly sampled pictures are manually labeled regarding (1) if they notably include human models; (2) the gender image from 'Female Exclusive',

[6] All products attached to the 'Men' or 'Women' categories are removed.

[7] https://www.faceplusplus.com/

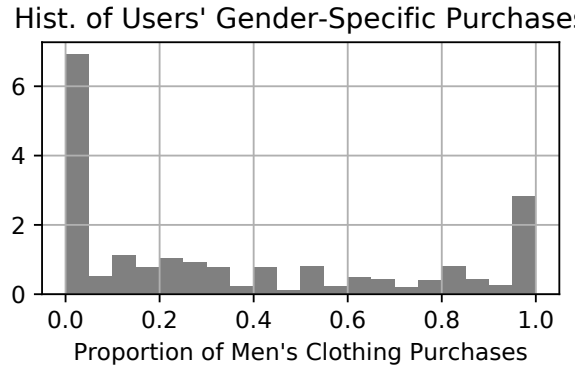


Figure 6.2: Distribution of purchase frequency towards gender-specific clothing products.

'Male Exclusive' or 'Both Famale & Male' (if multiple models are included in a single picture). We evaluate the human model detection results from the API based on these labels and find a high precision (96%) regarding the human model detection but a relatively low recall (53%). Note in our setting we are happy to discard ambiguous cases (sacrifice some recall) for the sake of high precision. We later randomly sample 100 products and manually decide if these products preserve any gender constraints based on their descriptions. Although 4 out of 100 products exhibit gender implications,[8] we don't find any strict constraints which prevent the unfavorable user identity group to consume these products.

User Identity Group (Gender)

We leverage users' interactions with amazon's *Clothing* products to access their gender identities, where most products are explicitly classified into *Women's Clothing* or *Men's Clothing*. As shown in Figure 6.2, we find a clear bimodel distribution of purchase frequency towards gender-specific clothing products. We discard 'ambiguous' users whose men's (or women's) clothing purchase frequency fall into 40%-60%, and identify the remaining users as 'Female' (69%) or 'Male' (31%). Finally 11% of total users in the *Electronics* category can be matched to these identities and 53% of them are identified as 'Female'.

[8]e.g. `https://www.amazon.com/gp/product/B00HX19EDI`

After removing products without any human models, we are still able to obtain a large-scale dataset containing around 1.3M rating scores across 9,560 electronics products from 1.1M users. Note that although the inferred product gender image and user identity are not as precise as in ***ModCloth***, this dataset is dramatically different from ***ModCloth*** regarding its scale and sparsity. In contrast to the relationship between a user's interactions with clothing products and a dimension of *human body shape*, we speculate that *gender* is intuitively less relevant to the intrinsic qualities of most products in ***Electronics***; thus its effects on users' interactions are possibly more likely to come from marketing bias.

6.4 Statistical Analysis

We split a consumer's product preference into two dimensions: (1) the user's preference on the willingness to consume (purchase) a product; and (2) the user's satisfaction feedback (e.g. ratings) on the consuming experience. We then conduct observational studies on the ***ModCloth*** and ***Electronics*** datasets to address marketing bias across the above two dimensions respectively.

- We first investigate if there is a bias introduced by a particular marketing effect in a consumer's *product selection* process. Specifically, we examine if a correlation exists between product image and user identity in terms of interaction frequency in our datasets.
- Then we study *consumer satisfaction* regarding the purchased products as a function of product image, user identity, and their second-order interactions. These consumer feedback signals include rating scores on ***ModCloth*** and ***Electronics***, as well as the binarized fit feedback (i.e., if the clothing product fits the user) on ***ModCloth***.

Table 6.2: Results from χ^2 test of the two-way contingency tables on ***ModCloth*** and ***Electronics***.

	ModCloth			Electronics		
	χ^2	p-value	#reviews	χ^2	p-value	#reviews
all	158.7	<0.001	91,526	581.8	<0.001	174,124
<=2014	0.5	0.466	25,383	151.0	<0.001	49,699
2015	66.7	<0.001	20,241	172.7	<0.001	46,891
2016	70.8	<0.001	21,239	96.4	<0.001	43,907
>=2017	29.0	<0.001	24,663	120.8	<0.001	33,627

6.4.1 Product Selection vs. Marketing Bias

Because of the constraint of conducting real-world experiments with random assignments, we instead address marketing bias in product selection by analyzing the association between product image and user identity in observed data with respect to interaction frequency. Our null hypothesis is that product image and user identity are statistically independent. Given this assumption, we expect to see lower deviations of their observed frequencies and the marginally expected values. Therefore the following Pearson's Chi-Squared Test Statistic can be used to test the association between these two variables in terms of frequency [48]:

$$\chi^2 = \sum_{m,n} \frac{(f_{m,n} - \mathbb{E}f_{m,n})^2}{\mathbb{E}f_{m,n}}, \tag{6.1}$$

where m and n represent a user identity group and a product image group respectively, $f_{m,n}$ is the observed number of interactions in the market segment (m,n) and $\mathbb{E}f_{m,n} = \frac{(\sum_{m'} f_{m',n})(\sum_{n'} f_{m,n'})}{(\sum_{m',n'} f_{m',n'})}$ represents its expectation. The null hypothesis will be rejected (i.e., the association between two variables exists in terms of frequency) if an extremely large χ^2 is obtained (i.e., small p-value).

To further separate the potential marketing bias from trending effects, we conduct association tests on the complete interaction data as well as interactions within different time spans. Test results are included in Table 6.2, where we find all p-values are smaller than 0.001 except for the test on interaction data before 2014 on ***ModCloth***. These results may imply the existence of the

Table 6.3: Contingency tables of the frequency distribution of product images and user identities on ***ModCloth*** and ***Electronics***. Deviations ($f_{m,n} - \mathbb{E}f_{m,n}$) from the expected frequency values are provided in parentheses.

(a) ModCloth

	User Identity		
Product Image	Small	Large	All
Small	31,800 (+754.98)	7,038 (-754.98)	38,838
Small&Large	41,361 (-754.98)	11,327 (+754.98)	52,688
All	73,161	18,365	91,526

(b) Electronics

	User Identity		
Product Image	Female	Male	All
Female	34,259 (+1,472.89)	31,587 (-1,472.89)	65,846
Female&Male	26,478 (+880.88)	24,930 (-880.88)	51,408
Male	25,963 (-2,353.77)	30,907 (+2,353.77)	56,870
All	86,700	87,424	174,124

association between product image and user identity in consumers' product selections.

In Table 6.3, we provide contingency tables of the frequency distribution of different market segments and their deviations from expected values ($f_{m,n} - \mathbb{E}f_{m,n}$). We observe generally more interactions than expected on the consumer-product segments where users' identities match the product images ('self-congruity'), while several market segments are underrepresented in the data. For example, ('Large' user, 'Small' product) on ***ModCloth*** and ('Female' user, 'Male' product) on ***Electronics*** have smaller market sizes compared with other market segments.

6.4.2 Consumer Satisfaction vs. Marketing Bias

Next we investigate consumer satisfaction as a function of product image and user identity through a standard statistical technique: two-way analysis of variance (**ANOVA**) [85]. We use rating scores to represent users' satisfaction regarding the overall quality of their consuming expe-

Table 6.4: Results from two-way analysis of variance (ANOVA) on ***ModCloth*** and ***Electronics***.

	ModCloth				Electronics	
	Rating		Fit		Rating	
	F-stat	p-value	F-stat	p-value	F-stat	p-value
product	200.9	<0.001	314.9	<0.001	369.4	<0.001
user	39.0	<0.001	376.0	<0.001	6.3	0.012
user:product	30.7	<0.001	0.0	0.997	0.9	0.404

rience on both ***ModCloth*** and ***Electronics***. For ***ModCloth***, we also study consumer satisfactions with respect to their fit feedback (where 'Just Right' is regarded as positive while all others are regarded as negative). The two-way **ANOVA** model can be formulated as

$$\textit{consumer satisfaction} \sim \textit{product} + \textit{user} + \textit{product} \times \textit{user},$$

where the null hypotheses of our tests include

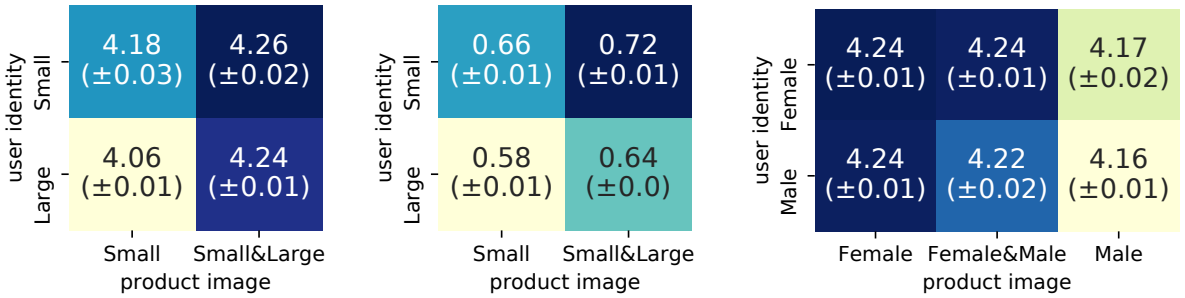
(a) the average consumer satisfaction is equal across different product image groups;

(b) the average consumer satisfaction is equal across different consumer identity groups;

(c) there is no interaction effect between product groups and consumer groups with respect to satisfaction.

Given these assumptions, we may expect a lower variance of average satisfactions across different groups (*between-group variation*) compared with the summation of satisfaction variations within each group (*within-group variation*). Therefore, the standard **F-statistic**, defined as the between-group variation divided by the within-group variation [85], can be applied to evaluate the correlations.

Results from statistical tests are included in Table 6.4. The heatmaps of sample means within market segments and their 95% confidence intervals are provided in Figure 6.3. We observe that users' rating scores are significantly different across market segments on ***ModCloth***. For example 'Large' users provide lower ratings on 'Small' products (Figure 6.3a). Although users'



(a) ModCloth (Rating) (b) ModCloth (Fit) (c) Electronics (Rating)

Figure 6.3: Heatmaps of sample means within market segments regarding (a) rating scores on ***ModCloth***, (b) fit feedback on ***ModCloth*** and (c) rating scores on ***Electronics***.

fit feedback differs across product groups and user groups (hypothesis (a) and (b) are rejected in Table 6.4), their association regarding fit feedback is negligible (results for 'user:product' in Table 6.4). According to Figure 6.3b, we find clothing products in the ***ModCloth*** dataset generally fit better on 'Small' users, and those products represented by human models with different body shapes ('Small&Large') tend to obtain better fit feedback. Although the 'self-congruity' pattern is significant in the product selection process on ***Electronics*** (see Table 6.2, Table 6.3b), the interaction between product 'gender' and user gender is insignificant with respect to users' rating scores (user:product in Table 6.4).

6.4.3 Summary of Observations

We summarize insights obtained from the above statistical analysis as follows. (1) The association between product image and user identity is consistently significant in terms of frequency distribution, implying the existence of marketing bias in the collected interaction datasets. The 'self-congruity' pattern is also observable, i.e., consumers may generally tend to interact with products with similar impressions as their identities. Such an association notably causes underrepresentations of certain market segments. (2) The relationship between consumer satisfaction and marketing factors is rather complicated. We observe rating disparities across

product groups and user groups, while the existence of their interaction effect depends on the type of product and the type of satisfaction measure. We find a similar 'self-congruity' pattern for rating scores on ***ModCloth*** while the 'user:product' term remains insignificant in the other two testing scenarios.

6.5 Market-Fairness of Recommender Systems

From the above analysis, we have confirmed that our interaction data is correlated to (and possibly affected by) marketing strategies used by product retailers (i.e., selections of human models). Our next step is to study if (and how) this marketing bias is propagated by algorithms from input data to recommendation results.

6.5.1 Problem Setting

In this study, we focus on recommendation algorithms trained on explicit feedback (i.e., rating scores). The primary predictive task is formulated as a *rating prediction problem*: rating scores ($r_{u,i}$) are assumed to reflect users' preferences over products, and algorithms are trained to generate users' product preference scores ($s_{u,i}$) which approximate these ratings.

Unlike previous studies [20, 53, 166, 182] which focus on evaluating and protecting the fairness of a single side (user or product) of recommender systems, in the context of marketing bias, we are particularly interested in the global market fairness of the recommendations, i.e., user-fairness and product-fairness need to be protected at the same time. Specifically we describe the market fairness in the explicit feedback setting along two dimensions. (1) Averaged errors of rating predictions from a recommendation algorithm across different consumer-product market segments are expected to be equal. (2) The distribution of market segments in terms of frequency within recommended interactions are expected to be consistent with the distribution within the real interaction data.

6.5.2 Rating Prediction Fairness

We notice that the first market fairness description indeed consistent with the null hypothesis of a one-way **ANOVA** test about the association between prediction errors ($e_{u,i} = s_{u,i} - r_{u,i}$) and market segments ($(m,n)$). That is, with the assumption that average prediction errors from a fair algorithm are supposed to be irrelevant to market segments, we expect to observe a lower variation of average errors across market segments (*bewteen-segment variation*) compared to the error variations within each segment (*within-segment variation*). Specifically these variations can be defined as

$$\begin{aligned} \text{between-segment var.:} \quad V^{(market)} &= \frac{1}{|D|} \sum_{m,n} |D_{m,n}| \Big(\bar{e}_{m,n,\cdot} - \bar{e} \Big)^2 \\ \text{within-segment var.:} \quad U^{(market)} &= \frac{1}{|D|} \sum_{m,n} \sum_{\substack{u \in U_m, \\ i \in I_n}} \Big(e_{u,i} - \bar{e}_{m,n,\cdot} \Big)^2 \end{aligned} \tag{6.2}$$

where $\bar{e}_{m,n,\cdot}$ denotes the sample mean of prediction errors within the market segment (m,n); $|D_{m,n}|$ represents the number of interactions included in a consumer-product segment (m,n); $|D|$ denotes the total sample size.

To ensure a tractable distribution for significance testing, the above two terms are corrected by their degrees-of-freedom and the following **F-statistic** can thus be calculated:

$$F^{(market)} = \frac{V^{(market)} \; / \; (M \times N - 1)}{U^{(market)} \; / \; \underbrace{(|D| - M \times N)}_{\text{correction of degree of freedom}}}. \tag{6.3}$$

Then we obtain a fairness evaluation metric to evaluate a global parity of prediction errors across different consumer-product market segments, where lower F indicates better rating prediction fairness.

6.5.3 Product Ranking Fairness

We further investigate the fairness of the product ranking performance from recommendation algorithms. For each user, we rank all products based on the predicted preference scores $s_{u,i}$ and regard the top-ranked K items as recommended products. By gathering users and the recommended products, we are able to obtain the frequency distribution of market segments within these predicted interactions $p_{m,n} \sim P$. We regard the frequency distribution of market segments in the real interactions $q_{m,n} \sim Q$ as the reference distribution, and evaluate the deviation of P from Q using the following **KL-divergence** [94]:

$$D_{KL}(P \mid Q) = \sum_{m,n} p_{m,n} \log\Big(\frac{p_{m,n}}{q_{m,n}}\Big). \tag{6.4}$$

We use this metric to evaluate the product ranking fairness. Lower D_{KL} indicates better fairness.

6.5.4 A Fairness-Aware Framework

A common optimization criterion for model-based collaborative filtering algorithm in the explicit feedback setting is based on **MSE**, i.e., minimizing the following loss function

$$\mathscr{L} = \sum (s_{u,i} - r_{u,i})^2. \tag{6.5}$$

A popular choice to model the preference score $s_{u,i}$ is through matrix factorization [90]

$$s_{u,i} = b_0 + b_i + b_u + \langle \boldsymbol{\gamma}_i, \boldsymbol{\gamma}_u \rangle . \tag{6.6}$$

In Eq. (6.6), b_0 is the global intercept, b_i and b_u are item-specific and user-specific offsets, $\boldsymbol{\gamma}_i$ and $\boldsymbol{\gamma}_u$ are d-dimensional embeddings to capture items' latent properties and users' latent preferences on these dimensions.

Error Correlation Loss

Following previous work using the regularizing schemes [4, 20, 166], we propose a fairness-aware framework by considering an error correlation loss to regularize systematic error biases on the market:

$$\mathscr{L}^* = \sum (s_{u,i} - r_{u,i})^2 + \alpha \mathscr{L}_{corr.}, \tag{6.7}$$

where $\mathscr{L}_{corr.}$ is an additional term to regularize the correlation between prediction errors $e_{u,i}$ and the distribution of market segments (m,n). α is a hyperparameter to control the trade-off between prediction accuracy and this correlation penalty term. In practice, we consider the following form by relaxing the evaluation metric Eq. (6.3):

$$\mathscr{L}_{corr.} = \kappa^{(u.)} \overbrace{\frac{V^{(u.)}}{U^{(u.)}}}^{\text{error parity on user identity}} + \kappa^{(p.)} \underbrace{\frac{V^{(p.)}}{U^{(p.)}}}_{\text{error parity on product image}} + \kappa^{(market)} \overbrace{\frac{V^{(market)}}{U^{(market)}}}^{\text{error parity on market segments}}, \tag{6.8}$$

where $V^{(u.)}, U^{(u.)}, V^{(p.)}, U^{(p.)}$ can be implemented by merging market segments within the same type of user identity groups or product image groups. $\kappa^{(u.)}, \kappa^{(p.)}, \kappa^{(market)} \in \{0,1\}^3$ are binary hyperparameters to instantiate different forms of correlation loss. For example, a selection of $(\kappa^{(u.)}, \kappa^{(p.)}, \kappa^{(market)}) = (1,0,0)$ represents that we only penalize the correlation between prediction errors and user identity groups.

6.6 Experiments

We conduct experiments on the collected ***ModCloth*** and ***Amazon*** datasets to evaluate the recommendation performance and the market fairness as described in Section 6.5.

The following standard algorithms are considered: (1) **itemCF**, an item-based collaborative filtering algorithm [100, 141]; (2) **userCF**, a user-centric collaborative filtering method

[68]; (3) **MF**, the matrix factorization method [90], where the value of preference prediction $s_{u,i}$ is unbounded; (4) **PoissonMF**, the hierarchical Bayesian framework where the preference factorization is linked to the rating score through a Poisson distribution, so that the preference score $s_{u,i}$ is bounded as a positive value [54]. By studying the recommendation outputs from these methods, we evaluate how standard collaborative filtering algorithms respond to the marketing bias in the input data.

We implement our proposed framework (**MF (corr.error)**), where $s_{u,i}$ is factorized using matrix factorization. By comparing its performance with the above methods (especially **MF**), we evaluate if the rating prediction and the product ranking fairness can be improved without losing much accuracy by adding the proposed correlation loss. Besides, we consider another two fairness-aware alternatives:

- **MF (corr.value)**, a method similar to **MF (corr.error)** except that $\mathscr{L}_{corr.}$ is implemented as the correlation between the predicted rating *values* $s_{u,i}$ and the market segments. By comparing **MF (corr.error)** with it, we evaluate the effectiveness of controlling the parity of prediction errors instead of the absolute statistical parity of prediction values.
- **MF (reweighted)**, a method where the loss function is reweighted based on the sizes of market segments in the training data. We also consider the following generic form of the loss function:

$$\mathscr{L} = \frac{\kappa^{(u.)}}{M}\sum_{m} MSE_m + \frac{\kappa^{(p.)}}{N}\sum_{n} MSE_n + \frac{\kappa^{(market)}}{MN}\sum_{m,n} MSE_{m,n}.$$

  By comparing it with other baselines, we study if the marketing bias can be alleviated by simply increasing the weights of underrepresented segments in the training data.

For all above methods, we evaluate their rating prediction accuracy through **MSE** and **MAE**, rating prediction fairness on **F-statistic** (Eq. (6.3)), product recommendation accuracy through **AUC** and **NDCG**, product ranking fairness on the **KL-divergence** (Eq. (6.4))

Experimental Details

We use the following rules to split interactions into train/validation/test sets: for users with at least two reviews, their most recent ratings are regarded as test set; for users with at least three reviews, their second-to-last ratings are used for validation; the remaining interactions are used for training. We apply the same analysis on both training and test sets as in Section 6.4, and find similar patterns except that less female users (40%) are included in the test set of ***Electronics***.[9]

We use the **ADAM** optimizer [84] with a learning rate of 0.001, a batch size of 512 and a fixed dimensionality of the latent embeddings in all model-based methods ($d = 10$). An ℓ_2 regularizor is applied on all model-based methods, where λ is selected from $\{0.01, 0.1, 1.0, 10\}$. The accuracy-fairness trade-off α is searched from $\{0.5, 1.0, 5.0, 10.0\}$. All these hyperparameters are selected based on the recommendation accuracy[10] on the validation set. For the fairness-aware methods, we search hyperparameters $\boldsymbol{\kappa} = (\boldsymbol{\kappa}^{(u.)}, \boldsymbol{\kappa}^{(p.)}, \boldsymbol{\kappa}^{(market)})$ from $\{(1,0,0), (0,1,0), (0,0,1), (1,1,0), (1,1,1)\}$. For each $\boldsymbol{\kappa}$, we first decide all other hyperparameters based on their recommendation accuracy, then select $\boldsymbol{\kappa}$ which yields the most fair recommendation results on the validation set. For each user, the top-10 ranked products are regarded as recommended items. Reviews in the test set where rating scores are larger than 3 are considered as reference interactions for the ranking task. All results are reported on the test set.

6.6.1 How does a standard collaborative filtering algorithm respond to biased input data?

We report the above mentioned rating prediction and product ranking metrics on ***ModCloth*** and ***Electronics***, regarding both accuracy and fairness, in Table 6.5. We first investigate the standard recommendation methods without any explicit fairness controls (i.e., **itemCF**, **userCF**, **PoissonMF** and **MF**). We observe that most methods yield biased prediction results on both

[9] 85% female users (vs. 73% male users) have only one review in our entire dataset.
[10] **MSE** for rating prediction accurcy and **NDCG** for product ranking.

Table 6.5: Recommendation results on ***ModCloth*** and ***Electronics***. The most accurate and the most fairest results are underlined.

(a) ModCloth

		Rating Prediction				Product Ranking		
	Method	MSE	MAE	F-stat	p-value	AUC	NDCG	KL
(a)	itemCF	1.398	0.841	2.568	0.053	0.601	0.121	0.557
	userCF	1.880	0.946	3.889	0.009	0.504	0.123	0.303
	PoissonMF	1.168	0.859	9.600	<0.001	0.638	0.151	0.001
	MF	1.176	0.859	9.805	<0.001	0.817	0.179	0.015
(b)	MF (reweighted)	1.290	0.872	8.402	<0.001	0.852	0.183	0.012
	MF (corr.value)	1.208	0.875	9.887	<0.001	0.549	0.123	0.484
	MF (corr.error)	1.204	0.873	1.667	0.172	0.818	0.179	0.003

(b) Electronics

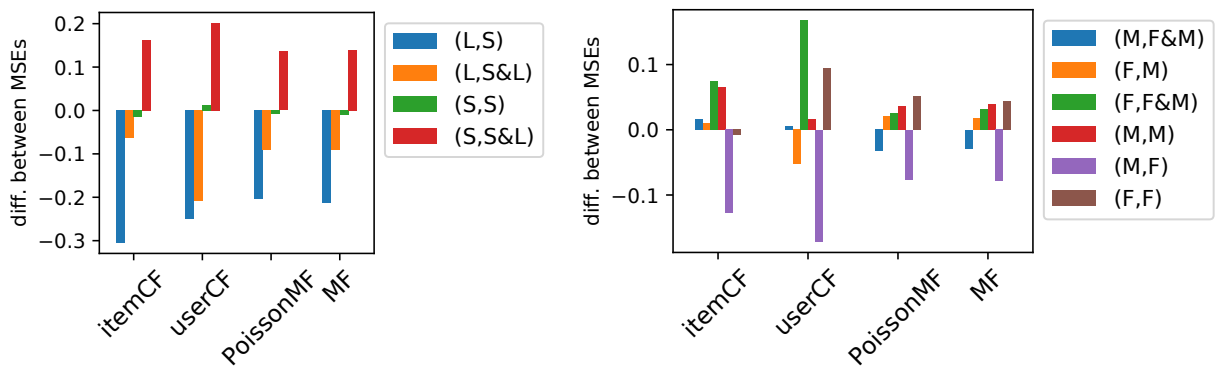
		Rating Prediction				Product Ranking		
	Method	MSE	MAE	F-stat	p-value	AUC	NDCG	KL
(a)	itemCF	1.529	0.966	5.099	<0.001	0.619	0.098	0.009
	userCF	2.487	0.980	1.501	0.186	0.503	0.087	0.009
	PoissonMF	1.628	1.035	4.112	0.001	0.565	0.085	0.014
	MF	1.590	1.025	3.447	0.004	0.591	0.091	0.012
(b)	MF (reweighted)	1.615	1.017	2.769	0.017	0.594	0.092	0.001
	MF (corr.value)	1.617	1.043	4.543	<0.001	0.502	0.086	0.012
	MF (corr.error)	1.543	1.011	1.896	0.091	0.766	0.122	0.002

datasets according to the **F-statistic**-based significance test. Although we find seemingly fair prediction errors from **userCF**, it actually produces a much larger **MSE** (as well as worse product ranking results) compared with other methods.

We further calculate the differences between the out-segment **MSE**s and the in-segment **MSE**s for these algorithms. Given a market segment (m,n), we have

$$diff_{m,n} = MSE_{u \notin U_m or i \notin I_n} - MSE_{u \in U_m and i \in I_n}. \tag{6.9}$$

$diff_{m,n} > 0$ indicates, for an algorithm, the market segment (m,n) is more predictable (smaller



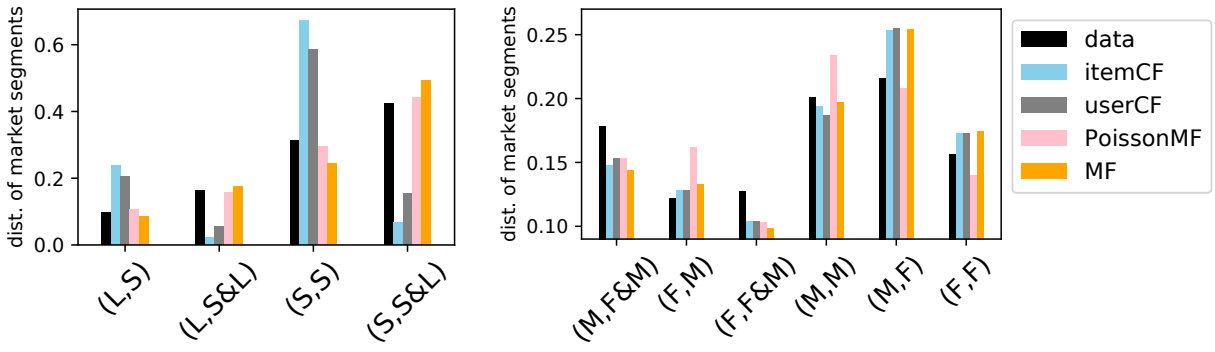
(a) ModCloth

(b) Electronics

Figure 6.4: Differences between the out-segment **MSE**s and the in-segment **MSE**s. Market segments are sorted based on their market sizes in the training data.

MSE) than the interactions outside it. These differences are displayed in Figure 6.4, where the market segments are sorted based on their training sizes. We observe an overall trend that all algorithms generally tend to favor the dominating market segments (e.g. 'Small' users on 'Small&Large' products in ***ModCloth***) in varying degrees. We find the correlation between the predictibility and the market segment size is more prominent on ***ModCloth*** but rather complicated on ***Electronics***. However, by cross matching Figure 6.4 and the contingency table Table 6.3b, we find that the trend correlates to the deviations of the real market size and the expected market size: the consumer-product segments ('Female', 'Male'), ('Male', 'Female') and ('Male', 'Female&Male') are underrepresented based on this difference ($f_{m,n} - \mathbb{E}f_{m,n} < 0$), also generally unfavored by the recommendation algorithms.

We display the distributions of market segments within real interaction data and the recommended top-10 products from these algorithms in Figure 6.5. Compared with the distributions in real interactions (the 'data' columns in Figure 6.5), we can observe the deviations of recommendation results from most algorithms, particularly **itemCF** and **userCF** on ***ModCloth***. However, systematic patterns about how these deviations correlate to the sizes of different market segments in the training data are not observed.



(a) ModCloth

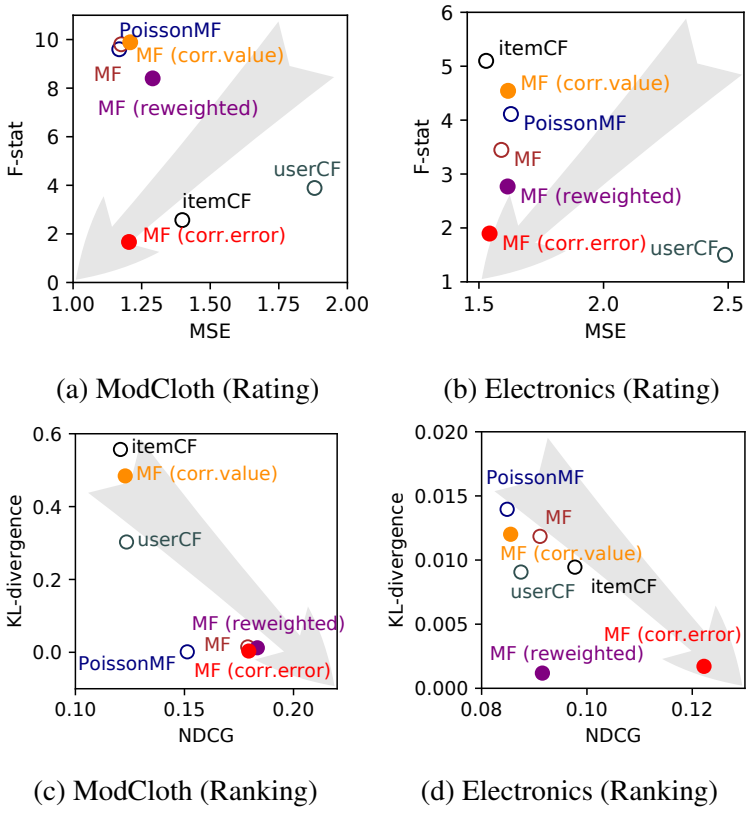
(b) Electronics

Figure 6.5: Distribution of market segments within test data and within recommendations. Market segments are sorted based on their sizes in training data.

6.6.2 Can recommendation fairness be improved by applying the correlation loss?

In Table 6.5, we further compare the results from fairness-aware algorithms in group (b) to the standard algorithms in group (a), particularly **MF**. To better visualize the trade-off between the recommendation accuracy and the market fairness, we present the scatter plots of an accuracy metric and a fairness metric on both datasets in Figure 6.6. We notice the proposed method with error correlation loss **MF (corr.error)** generally provides a better rating and ranking fairness (lower **F-statistic** and **KL-divergence**) than a standard **MF**, without trading-off much recommendation accuracy. An interesting finding is the combination selection $\boldsymbol{\kappa}$ on the validation set is consistent with our analysis in Table 6.4: the complete correlation loss ($\boldsymbol{\kappa} = (1,1,1)$) is selected for ***ModCloth*** and the addition of product and user correlation ($\boldsymbol{\kappa} = (1,1,0)$) is selected for ***Electronics***.

We find the reweighting scheme also benefits the fairness metrics, particularly in the product ranking setting. One surprising finding is by applying the error correlation loss, a significant performance gain in terms of product ranking accuracy (**AUC** and **NDCG**) can be obtained on ***Electronics***. A possible reason could be ***Electronics*** is an extremely sparse dataset



(a) ModCloth (Rating) (b) Electronics (Rating)

(c) ModCloth (Ranking) (d) Electronics (Ranking)

Figure 6.6: Scatter plots for accuracy-fairness trade-off from different algorithms. Shaded arrows indicate the most ideal direction: higher accuracy, better fairness.

where algorithms like **MF** may struggle converging to an ideal local optimum. The fairness-aware correlation loss, however, could help regularize the training process.

6.7 Conclusions and Future Work

We conclude our work and summarize the insights as follows. (1) We investigated a potential source of bias—marketing bias—in the form of the association between interaction feedback, product image and user identity, on two real-world e-commerce datasets. Through observational studies, the inter-correlations between these factors can be confirmed and the 'self-congruity' patterns are noticeable in the product selection process, which eventually results in the underrepresentation of some market segments. (2) We focused on the market fairness and

investigated how standard collaborative filtering algorithms reacted to this biased input data. We found such a bias can be propagated to the recommendation outcomes in varying degrees. (3) We developed an error correlation framework, which explicitly calibrates the equity of prediction errors across different market segments. Experimental results demonstrate that by applying this correlation loss, a superior accuracy-fairness trade-off can be achieved.

This work is a first step to approach the potential marketing bias in machine learning systems. We also wish to address several limitations of our data and methods, and provide the potential research directions.

- **Data.** We study the marketing bias by formulating it as the relationship between the human model images of products and users identities. Multiple marketing factors (e.g. product descriptions, social media advertisement contents) can also be considered. Binary gender identities are inferred in our ***Electronics*** dataset, which is limited to represent user identities that are not exclusively masculine or feminine, e.g. users who don't always purchase products corresponding to their own identities, or those who identify themselves outside the binary definition.
- **Analysis.** We collect ***ModCloth*** and ***Electronics*** as logged interactions where many confounders (inventory status, the observability of each product, etc.) exist and are difficult to be disentangled. Although the inter-correlation between product image and user identity is observed in these datasets, we cannot draw any causal conclusions without controlling some notable confounding factors. Therefore another direction to validate (or more fundamentally address) this marketing bias is to conduct randomized experiments or natural experiments. In this way, causal conclusions and insights can be provided to product sellers and recommender system practitioners.
- **Algorithm.** Although we only focus on algorithms trained on explicit feedback, it is relatively intuitive to extend the proposed error correlation framework to other pointwise recommendation algorithms. Another direction is to address the marketing bias in pairwise

ranking recommendation algorithms, where new market fairness metrics and debiasing methods can be considered.

6.8 Acknowledgements

This chapter contains the material to appear in *ACM Conference on Web Search and Data Mining*, 2020 ("Addressing Marketing Bias in Product Recommendations," Mengting Wan, Jianmo Ni, Rishabh Misra, and Julian McAuley). The dissertation author was the primary investigator and author of this paper.

Chapter 7

Conclusions

In this dissertation, I have presented my research for modeling the dynamics of consumer behavior from massive and heterogeneous interaction data. We revisited the assumptions, the objectives, and potential societal concerns of ML-based recommendation algorithms, by considering several pronounced economic factors, possible structures of different types of consumer activities, and the potential marketing biases in the interaction data. We pointed out a new direction where unstructured texts could be a useful resource to model consumer behavior, though via relatively classic NLP and information retrieval (IR) techniques. With the great success of recent language models such as **ELMo** [131], **BERT** [42], **GPT** [135, 136] and **XLNet** [165], modeling consumer behavior with unstructured textual feedback becomes an even more exciting research direction. In this regard, I believe the introduction of well-developed economic behavioral theories and advanced language models into recommender systems (and related applications) could be beneficial for multiple fields. My research hopefully could contribute to this interdisciplinary conversation.

Bibliography

[1] The instacart online grocery shopping dataset 2017. Accessed on Dec. 2017. https://www.instacart.com/datasets/grocery-shopping-2017.

[2] Supermarket Facts. Food Marketing Institute, Accessed Oct. 2016. http://www.fmi.org/research-resources/supermarket-facts.

[3] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[4] H. Abdollahpouri, R. Burke, and B. Mobasher. Controlling popularity bias in learning-to-rank recommendation. In *RecSys*, 2017.

[5] D. A. Ackerberg. Advertising, learning, and consumer choice in experience good markets: an empirical examination. *International Economic Review*, 44(3):1007–1040, 2003.

[6] D. Agarwal and B.-C. Chen. Regression-based latent factor models. In *KDD*, 2009.

[7] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In *SIGMOD*, 1993.

[8] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *VLDB*, 1994.

[9] A. Ahmed, B. Kanagal, S. Pandey, V. Josifovski, L. G. Pueyo, and J. Yuan. Latent factor models with additive and hierarchically-smoothed user preferences. In *WSDM*, 2013.

[10] A. Anderson, R. Kumar, A. Tomkins, and S. Vassilvitskii. The dynamics of repeat consumption. In *WWW*, 2014.

[11] N. Arora, G. M. Allenby, and J. L. Ginter. A hierarchical bayes model of primary and secondary demand. *Marketing Science*, 17(1):29–44, 1998.

[12] S. Athey and G. W. Imbens. Machine learning methods that economists should know about. *Annual Review of Economics*, 11, 2019.

[13] A. Balahur, E. Boldrini, A. Montoyo, and P. Martinez-Barco. Opinion and generic question answering systems: a performance analysis. In *ACL-IJCNLP*, 2009.

[14] A. Balahur, E. Boldrini, A. Montoyo, and P. Martínez-Barco. Going beyond traditional QA systems: challenges and keys in opinion question answering. In *COLING*, 2010.

[15] A. Balahur, E. Boldrini, A. Montoyo, and P. Martínez-Barco. Opinion question answering: Towards a unified approach. In *ECAI*, 2010.

[16] L. Baltrunas, B. Ludwig, and F. Ricci. Matrix factorization techniques for context aware recommendation. In *RecSys*, 2011.

[17] O. Barkan and N. Koenigstein. Item2vec: Neural item embedding for collaborative filtering. In *Workshop on Machine Learning for Signal Processing*, 2016.

[18] K. Bauman, B. Liu, and A. Tuzhilin. Aspect based recommendations: Recommending items with the most valuable aspects based on user reviews. In *KDD*, 2017.

[19] A. R. Benson, R. Kumar, and A. Tomkins. Modeling user consumption sequences. In *WWW*, 2016.

[20] A. Beutel, J. Chen, T. Doshi, H. Qian, L. Wei, Y. Wu, L. Heldt, Z. Zhao, L. Hong, E. H. Chi, et al. Fairness in recommendation ranking through pairwise comparisons. In *KDD*, 2019.

[21] A. E. Birdwell. A study of the influence of image congruence on consumer choice. *The Journal of Business*, 41(1):76–88, 1968.

[22] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *JMLR*, 2003.

[23] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *COMPSTAT*, pages 177–186, 2010.

[24] R. Bro. Parafac. tutorial and applications. *Chemometrics and intelligent laboratory systems*, 1997.

[25] R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.

[26] R. Burke, N. Sonboli, and A. Ordonez-Gauger. Balanced neighborhoods for multi-sided fairness in recommendation. In *FAT**, 2018.

[27] R. Carbaugh. *Contemporary economics: an applications approach*. Routledge, 2016.

[28] R. Catherine and W. W. Cohen. Transnets: Learning to transform for recommendation. In *RecSys*, 2017.

[29] J. Chen, C. Wang, and J. Wang. Modeling the interest-forgetting curve for music recommendation. In *MM*, 2014.

[30] J. Chen, C. Wang, and J. Wang. A personalized interest-forgetting markov model for recommendations. In *AAAI*, 2015.

[31] T. Chen, W. Zhang, Q. Lu, K. Chen, Z. Zheng, and Y. Yu. SVDFeature: a toolkit for feature-based collaborative filtering. *JMLR*, 13:3619–3622, 2012.

[32] H. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, and H. Shah. Wide & deep learning for recommender systems. In *DLRS Workshop@RecSys*, 2016.

[33] P. K. Chintagunta. Investigating purchase incidence, brand choice and purchase quantity decisions of households. *Marketing Science*, 12(2):184–208, 1993.

[34] L. G. Cooper and M. Nakanishi. *Market-share analysis: Evaluating competitive marketing effectiveness*, volume 1. Springer Science & Business Media, 1989.

[35] P. Covington, J. Adams, and E. Sargin. Deep neural networks for youtube recommendations. In *RecSys*, 2016.

[36] P. Covington, J. Adams, and E. Sargin. Deep neural networks for youtube recommendations. In *RecSys*, 2016.

[37] P. J. Danaher, M. S. Smith, K. Ranasinghe, and T. S. Danaher. Where, when, and how long: factors that influence the redemption of mobile phone coupons. *Journal of Marketing Research*, 52(5):710–725, 2015.

[38] J. Dawes, L. Meyer-Waarden, and C. Driesener. Has brand loyalty declined? a longitudinal analysis of repeat purchase behavior in the uk and the usa. *Journal of Business Research*, 68(2):425–432, 2015.

[39] A. Deaton and J. Muellbauer. An almost ideal demand system. *The American economic review*, 70(3):312–326, 1980.

[40] A. M. Degeratu, A. Rangaswamy, and J. Wu. Consumer choice behavior in online and traditional supermarkets: The effects of brand name, price, and other search attributes. *International Journal of research in Marketing*, 17(1):55–78, 2000.

[41] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society*, 1977.

[42] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[43] Q. Diao, M. Qiu, C. Wu, A. J. Smola, J. Jiang, and C. Wang. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In *KDD*, 2014.

[44] Y. Dong, N. V. Chawla, and A. Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *KDD*, 2017.

[45] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 2011.

[46] J. Eisenstein, A. Ahmed, and E. P. Xing. Sparse additive generative models of text. In *ICML*, 2011.

[47] M. D. Ekstrand, M. Tian, M. R. I. Kazi, H. Mehrpouyan, and D. Kluver. Exploring author gender in book rating and recommendation. In *RecSys*, 2018.

[48] B. S. Everitt. *The analysis of contingency tables*. Chapman and Hall/CRC, 1992.

[49] T. Fawcett. An introduction to ROC analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[50] P. Forbes and M. Zhu. Content-boosted matrix factorization for recommender systems: experiments with recipe recommendation. In *RecSys*, 2011.

[51] S. C. Gadanho and N. Lhuillier. Addressing uncertainty in implicit preferences. In *RecSys*, 2007.

[52] H. Gao, J. Tang, X. Hu, and H. Liu. Content-aware point of interest recommendation on location-based social networks. In *AAAI*, 2015.

[53] S. Garg, V. Perot, N. Limtiaco, A. Taly, E. H. Chi, and A. Beutel. Counterfactual fairness in text classification through robustness. In *AIES*, 2019.

[54] P. Gopalan, J. M. Hofman, and D. M. Blei. Scalable recommendation with hierarchical poisson factorization. In *UAI*, 2015.

[55] S. L. Grau and Y. C. Zotos. Gender stereotypes in advertising: a review of current research. *International Journal of Advertising*, 35(5):761–770, 2016.

[56] M. Grbovic, V. Radosavljevic, N. Djuric, N. Bhamidipati, J. Savla, V. Bhagwan, and D. Sharp. E-commerce in your inbox: Product recommendations at scale. In *KDD*, 2015.

[57] E. L. Grubb and H. L. Grathwohl. Consumer self-concept, symbolism and market behavior: A theoretical approach. *Journal of Marketing*, 31(4):22–27, 1967.

[58] E. L. Grubb and G. Hupp. Perception of self, generalized stereotypes, and brand selection. *Journal of Marketing research*, 5(1):58–63, 1968.

[59] A. Gunawardana and C. Meek. A unified approach to building hybrid recommender systems. In *RecSys*, 2009.

[60] S. Gupta. Impact of sales promotions on when, what, and how much to buy. *Journal of Marketing research*, pages 342–355, 1988.

[61] T. Gurbanov and F. Ricci. Action prediction models for recommender systems based on collaborative filtering and sequence mining hybridization. In *Proceedings of the Symposium on Applied Computing*, 2017.

[62] J. He and D. Dai. Summarization of yes/no questions using a feature function model. *JMLR*, 2011.

[63] J. He and D. Dai. Summarization of yes/no questions using a feature function model. In *ACML*, 2011.

[64] R. He, W.-C. Kang, and J. McAuley. Translation-based recommendation. In *RecSys*, 2017.

[65] R. He and J. McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *WWW*, 2016.

[66] R. He and J. McAuley. Vbpr: Visual bayesian personalized ranking from implicit feedback. In *AAAI*, 2016.

[67] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua. Neural collaborative filtering. In *WWW*, 2017.

[68] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *SIGIR*, 1999.

[69] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk. Session-based recommendations with recurrent neural networks. In *ICLR*, 2015.

[70] C.-K. Hsieh, L. Yang, Y. Cui, T.-Y. Lin, S. Belongie, and D. Estrin. Collaborative metric learning. In *WWW*, 2017.

[71] M. Hu and B. Liu. Mining and summarizing customer reviews. In *KDD*, 2004.

[72] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, 2008.

[73] J. Jacoby and D. B. Kyner. Brand loyalty vs. repeat purchasing behavior. *Journal of Marketing research*, pages 1–9, 1973.

[74] D. Jannach, L. Lerche, I. Kamehkhosh, and M. Jugovac. What recommenders recommend: an analysis of recommendation biases and possible countermeasures. *User Modeling and User-Adapted Interaction*, 25(5):427–491, 2015.

[75] G. Jawaheer, M. Szomszor, and P. Kostkova. Comparison of implicit and explicit feedback from an online music recommendation service. In *HetRec*, 2010.

[76] G. Jawaheer, P. Weller, and P. Kostkova. Modeling user preferences in recommender systems: A classification framework for explicit and implicit user feedback. *ACM Transactions on Interactive Intelligent Systems*, 2014.

[77] Y. Jiang and Y. Liu. Optimization of online promotion: a profit-maximizing model integrating price discount and product recommendation. *International Journal of Information Technology & Decision Making*, 11(05):961–982, 2012.

[78] Y. Jiang, J. Shang, Y. Liu, and J. May. Redesigning promotion strategy for e-commerce competitiveness through pricing and recommendation. *International Journal of Production Economics*, 167:257–270, 2015.

[79] T. Joachims, A. Swaminathan, and T. Schnabel. Unbiased learning-to-rank with biased feedback. In *WSDM*, 2017.

[80] C. C. Johnson. Logistic matrix factorization for implicit feedback data. *NeurIPS*, 2014.

[81] K. Jones, S. Walker, and S. Robertson. A probabilistic model of information retrieval: development and comparative experiments. *Information Processing & Management*, 2000.

[82] M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 1994.

[83] K. Kapoor, K. Subbian, J. Srivastava, and P. Schrater. Just in time recommendations: Modeling the dynamics of boredom in activity streams. In *WSDM*, 2015.

[84] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[85] D. G. Kleinbaum, L. L. Kupper, K. E. Muller, and A. Nizam. *Applied regression analysis and other multivariable methods*, volume 601. Duxbury Press Belmont, CA, 1988.

[86] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD*, 2008.

[87] Y. Koren. Collaborative filtering with temporal dynamics. In *KDD*, 2009.

[88] Y. Koren. Factor in the neighbors: Scalable and accurate collaborative filtering. *TKDD*, 2010.

[89] Y. Koren and R. Bell. Advances in collaborative filtering. In *Recommender Systems Handbook*. Springer, 2011.

[90] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

[91] D. Kotzias, M. Lichman, and P. Smyth. Predicting consumption patterns with repeated and novel events. *TKDE*, 2018.

[92] F. Kressmann, M. J. Sirgy, A. Herrmann, F. Huber, S. Huber, and D.-J. Lee. Direct and indirect effects of self-image congruence on brand loyalty. *Journal of Business research*, 59(9):955–964, 2006.

[93] L.-W. Ku, Y.-T. Liang, and H.-H. Chen. Question analysis and answer passage retrieval for opinion question answering systems. In *ROCLING*, 2007.

[94] S. Kullback and R. A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[95] T. Lang and M. Rettenmeier. Understanding consumer behavior with recurrent neural networks. In *Workshop on Machine Learning Methods for Recommender Systems*, 2017.

[96] D. T. Le, H. W. Lauw, and Y. Fang. Basket-sensitive personalized item recommendation. In *IJCAI*, 2017.

[97] L. Lerche, D. Jannach, and M. Ludewig. On the value of reminders within e-commerce recommendations. In *UMAP*, 2016.

[98] F. Li, Y. Tang, M. Huang, and X. Zhu. Answering opinion questions with random walks on graphs. In *ACL-IJCNLP*, 2009.

[99] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *ACL Workshop on Text Summarization Branches Out*, 2004.

[100] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, (1):76–80, 2003.

[101] G. Ling, M. R. Lyu, and I. King. Ratings meet reviews, a combined approach to recommend. In *RecSys*, 2014.

[102] J. Liu, G. Wu, and J. Yao. Opinion searching in multi-product reviews. In *International Conference on Computer and Information Technology*, 2006.

[103] N. N. Liu, E. W. Xiang, M. Zhao, and Q. Yang. Unifying explicit and implicit feedback for collaborative filtering. In *CIKM*, 2010.

[104] N. T. Longford. Random coefficient models. In *Handbook of statistical modeling for the social and behavioral sciences*, pages 519–570. Springer, 1995.

[105] P. Lops, M. De Gemmis, and G. Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*, pages 73–105. Springer, 2011.

[106] L. v. d. Maaten and G. Hinton. Visualizing data using t-SNE. *JMLR*, 2008.

[107] N. K. Malhotra. Self concept and product choice: An integrated perspective. *Journal of Economic Psychology*, 9(1):1–28, 1988.

[108] N. Mankiw. *Principles of Microeconomics*. Economics Series. Cengage Learning, 2011.

[109] C. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Cambridge university press, 2008.

[110] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *ACL System Demo*, 2014.

[111] J. McAuley and J. Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *RecSys*, 2013.

[112] J. McAuley, C. Targett, Q. Shi, and A. Van Den Hengel. Image-based recommendations on styles and substitutes. In *SIGIR*, 2015.

[113] J. McAuley and A. Yang. Addressing complex and subjective product-related queries with customer reviews. In *WWW*, 2016.

[114] C. E. McCulloch and J. M. Neuhaus. *Generalized linear mixed models*. Wiley Online Library, 2001.

[115] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *ICLR workshop*, 2013.

[116] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, 2013.

[117] R. Misra, M. Wan, and J. McAuley. Decomposing fit semantics for product size recommendation in metric spaces. In *RecSys*, 2018.

[118] S. Moghaddam and M. Ester. AQA: aspect-based opinion question answering. In *ICDMW*, 2011.

[119] F. J. Molnar, B. Hutton, and D. Fergusson. Does analysis using "last observation carried forward" introduce bias in dementia research? *Canadian Medical Association Journal*, 179(8):751–753, 2008.

[120] J. Ni, Z. C. Lipton, S. Vikram, and J. J. McAuley. Estimating reactions and recommending products with generative models of reviews. In *IJCNLP*, 2017.

[121] X. Ning and G. Karypis. Sparse linear methods with side information for top-n recommendations. In *RecSys*, 2012.

[122] J. Nocedal and S. Wright. *Numerical optimization*. Springer, 2006.

[123] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang. One-class collaborative filtering. In *ICDM*, 2008.

[124] W. Pan, N. N. Liu, E. W. Xiang, and Q. Yang. Transfer learning to predict missing ratings via heterogeneous user feedbacks. In *IJCAI*, 2011.

[125] S. Park, Y.-D. Kim, and S. Choi. Hierarchical bayesian matrix factorization with side information. In *IJCAI*, 2013.

[126] D. Parra and X. Amatriain. Walk the talk: Analyzing the relation between implicit and explicit feedback for preference elicitation. In *UMAP*, 2011.

[127] D. Parra, A. Karatzoglou, X. Amatriain, and I. Yavuz. Implicit feedback recommendation via implicit-to-explicit ordinal logistic regression mapping. In *CARS*, 2011.

[128] A. Pathak, K. Gupta, and J. McAuley. Generating and personalizing bundle recommendations on steam. In *SIGIR*, 2017.

[129] M. J. Pazzani and D. Billsus. Content-based recommendation systems. In *The Adaptive Web*, pages 325–341. Springer, 2007.

[130] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

[131] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *NAACL-HLT*, 2018.

[132] I. Porteous, A. U. Asuncion, and M. Welling. Bayesian matrix factorization with side information and dirichlet process mixtures. In *AAAI*, 2010.

[133] I. Porteous, D. Newman, A. Ihler, A. Asuncion, P. Smyth, and M. Welling. Fast collapsed gibbs sampling for latent dirichlet allocation. In *KDD*, 2008.

[134] P. Quester and A. Lin Lim. Product involvement/brand loyalty: is there a link? *Journal of product & brand management*, 12(1):22–38, 2003.

[135] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. *OpenAI Blog*, 2018.

[136] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019.

[137] V. Raykar, S. Yu, L. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. *JMLR*, 2010.

[138] S. Rendle. Factorization machines with libfm. *TIST*, 3(3):57, 2012.

[139] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, 2009.

[140] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *WWW*, 2010.

[141] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW*, 2001.

[142] J. B. Schafer, J. Konstan, and J. Riedl. Recommender systems in e-commerce. In *EC*, 1999.

[143] T. Schnabel, A. Swaminathan, A. Singh, N. Chandak, and T. Joachims. Recommendations as treatments: Debiasing learning and evaluation. In *ICML*, 2016.

[144] S. Seo, J. Huang, H. Yang, and Y. Liu. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *RecSys*, 2017.

[145] A. P. Singh and G. J. Gordon. A unified view of matrix factorization models. In *ECML-PKDD*, 2008.

[146] M. J. Sirgy. Self-concept in consumer behavior: A critical review. *Journal of consumer research*, 9(3):287–300, 1982.

[147] M. J. Sirgy, D. Grewal, T. F. Mangleburg, J.-o. Park, K.-S. Chon, C. B. Claiborne, J. S. Johar, and H. Berkman. Assessing the predictive validity of two methods of measuring self-image congruence. *Journal of the academy of marketing science*, 25(3):229, 1997.

[148] V. Stoyanov, C. Cardie, and J. Wiebe. Multi-perspective question answering using the OpQA corpus. In *EMNLP*, 2005.

[149] Z. Sun, J. Yang, J. Zhang, A. Bozzon, Y. Chen, and C. Xu. MRLR: multi-level representation learning for personalized ranking in recommendation. In *IJCAI*, 2017.

[150] Y. Tay, A. T. Luu, and S. C. Hui. Multi-pointer co-attention networks for recommendation. In *KDD*, 2018.

[151] Y. W. Teh, D. Newman, and M. Welling. A collapsed variational bayesian inference algorithm for latent dirichlet allocation. In *NeurIPS*, 2006.

[152] P. Umberto. Developing a price-sensitive recommender system to improve accuracy and business performance of ecommerce applications. *International Journal of Electronic Commerce Studies*, 6(1):1, 2015.

[153] F. Vasile, E. Smirnova, and A. Conneau. Meta-prod2vec: Product embeddings using side-information for recommendation. In *RecSys*, 2016.

[154] M. Wan and J. J. McAuley. Item recommendation on monotonic behavior chains. In *RecSys*, 2018.

[155] C. Wang and D. M. Blei. Collaborative topic modeling for recommending scientific articles. In *KDD*, 2011.

[156] H. Wang, Y. Lu, and C. Zhai. Latent aspect rating analysis on review text data: a rating regression approach. In *KDD*, 2010.

[157] H. Wang, N. Wang, and D. Yeung. Collaborative deep learning for recommender systems. In *KDD*, 2015.

[158] J. Wang, P. Huang, H. Zhao, Z. Zhang, B. Zhao, and D. L. Lee. Billion-scale commodity embedding for e-commerce recommendation in alibaba. In *KDD*, 2018.

[159] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, and X. Cheng. Learning hierarchical representation model for next-basket recommendation. In *SIGIR*, 2015.

[160] X. Wang, X. He, M. Wang, F. Feng, and T. Chua. Neural graph collaborative filtering. In *SIGIR*, 2019.

[161] J. Weston, S. Bengio, and N. Usunier. Large scale image annotation: learning to rank with joint word-image embeddings. *Machine learning*, 2010.

[162] C. Wu and M. Yan. Session-aware information embedding for e-commerce product recommendation. In *CIKM*, 2017.

[163] Y. Wu and M. Ester. Flame: A probabilistic model combining aspect based opinion mining and collaborative filtering. In *WSDM*, 2015.

[164] L. Yang, Y. Cui, Y. Xuan, C. Wang, S. Belongie, and D. Estrin. Unbiased offline recommender evaluation for missing-not-at-random implicit feedback. In *RecSys*, 2018.

[165] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*, 2019.

[166] S. Yao and B. Huang. Beyond parity: Fairness objectives for collaborative filtering. In *NeurIPS*, 2017.

[167] W. Yao, J. He, H. Wang, Y. Zhang, and J. Cao. Collaborative topic ranking: Leveraging item meta-data for sparsity reduction. In *AAAI*, 2015.

[168] H. Yin, H. Chen, X. Sun, H. Wang, Y. Wang, and Q. V. H. Nguyen. Sptf: A scalable probabilistic tensor factorization model for semantic-aware behavior prediction. In *ICDM*, 2017.

[169] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *KDD*, 2018.

[170] H. Yu and V. Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *EMNLP*, 2003.

[171] J. Yu, Z.-J. Zha, and T.-S. Chua. Answering opinion questions on products by exploiting hierarchical organization of consumer reviews. In *EMNLP-CoNLL*, 2012.

[172] J. Zhang and L. Krishnamurthi. Customizing promotions in online stores. *Marketing Science*, 23(4):561–578, 2004.

[173] J. Zhang and M. Wedel. The effectiveness of customized promotions in online and offline stores. *Journal of Marketing Research*, 46(2):190–206, 2009.

[174] X. Zhang, J. Zhao, and J. Lui. Modeling the assimilation-contrast effects in online product rating systems: Debiasing and recommendations. In *RecSys*, 2017.

[175] X. Zhang, Y. Zhou, Y. Ma, B. Chen, L. Zhang, and D. Agarwal. Glmix: Generalized linear mixed models for large-scale response prediction. In *KDD*, 2016.

[176] Y. Zhang, Q. Ai, X. Chen, and B. Croft. Joint representation learning for top-n recommendation with heterogeneous information sources. In *CIKM*, 2017.

[177] T. Zhao, J. McAuley, and I. King. Leveraging social connections to improve personalized ranking for collaborative filtering. In *CIKM*, 2014.

[178] L. Zheng, C.-T. Lu, F. Jiang, J. Zhang, and P. S. Yu. Spectral collaborative filtering. In *RecSys*, 2018.

[179] L. Zheng, V. Noroozi, and P. S. Yu. Joint deep modeling of users and items using reviews for recommendation. In *WSDM*, 2017.

[180] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, and K. Gai. Deep interest network for click-through rate prediction. In *KDD*, 2018.

[181] M. Zhou, Z. Ding, J. Tang, and D. Yin. Micro behaviors: A new perspective in e-commerce recommender systems. In *WSDM*, 2018.

[182] Z. Zhu, X. Hu, and J. Caverlee. Fairness-aware tensor-based recommendation. In *CIKM*, 2018.